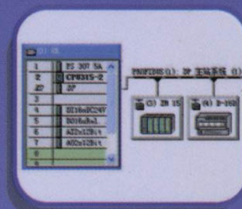
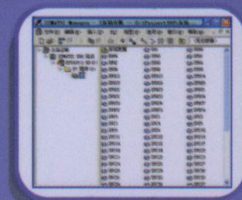
跟我动手学

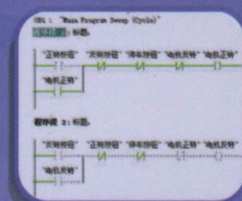
S7-300/400 PLC

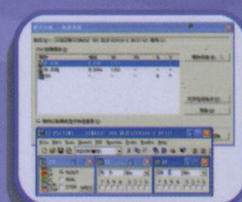
廖常初 主编













- 使用随书光盘的编程软件和仿真软件做完书中的四十多个实训，就能快速掌握S7-300/400的使用方法
- 随书光盘还附有西门子人机界面的组态软件WinCCflexible 2008、视频教程、三十多个例程和用户手册





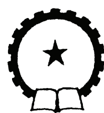
机械工业出版社
CHINA MACHINE PRESS

VISIT…

LANZAROTE Caliente.com

跟我动手学 S7-300/400 PLC

廖常初　主编



机 械 工 业 出 版 社

S7-300/400 PLC 是国内应用最广、市场占有率最高的大中型 PLC。本书强调通过实际操作进行学习。书中有四十多个实训，随书光盘有三十多个与正文配套的例程和多媒体视频教程。读者一边看书，一边根据实训的要求，用随书光盘中的编程软件和仿真软件在计算机上做仿真实验，就能快速掌握软件安装、硬件和网络组态、编程、监控、故障诊断、指令应用、程序结构、编程方法、通信、人机界面应用和 PID 控制等方面的知识和操作方法。PID 闭环仿真例程可以用来学习整定 PID 参数的方法。

本书可以供工程技术人员自学，也可以作为高职高专、技工学校有关专业的教材。

图书在版编目（CIP）数据

跟我动手学 S7-300/400 PLC / 廖常初主编. —北京：机械工业出版社，2010.9（2015.6 重印）

ISBN 978-7-111-31641-1

Ⅰ. ①跟… Ⅱ. ①廖… Ⅲ. ①可编程序控制器 Ⅳ. ①TM571.6

中国版本图书馆 CIP 数据核字（2010）第 163372 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策划编辑：时 静 责任编辑：时 静
责任印制：乔 宇
保定市中画美凯印刷有限公司印刷
2015 年 6 月第 1 版・第 6 次印刷
184mm×260mm・14 印张・343 千字
24001－28000 册
标准书号：ISBN 978-7-111-31641-1
ISBN 978-7-89451-660-2（光盘）
定价：32.00 元（含 1DVD）

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务
社服务中心：（010）88361066
销售一部：（010）68326294
销售二部：（010）88379649
读者购书热线：（010）88379203

网络服务
门户网：http://www.cmpbook.com
教材网：http://www.cmpedu.com
封面无防伪标均为盗版

前　言

S7-300/400 PLC 是国内应用最广、市场占有率最高的大中型 PLC。很多同行都觉得 S7-300/400 不容易入门，自学非常困难。S7-PLCSIM 是 S7-300/400 功能强大、使用方便的仿真软件，是学习 S7-300/400 的理想工具。可以用它在通用计算机上做仿真实验，模拟 PLC 硬件的运行和执行用户程序。仿真实验和做硬件实验时观察到的现象几乎完全一样。

看十遍书不如动一次手，本书的特点是强调通过实际操作来学习。本书提供了四十多个精心设计的实训，随书光盘有三十多个与正文配套的例程，绝大多数实训都可以做仿真实验。专门设计的实训使应用中的难点有关的知识形象化，易于理解。对于操作中的重点和难点，还提供了配套的多媒体视频教程。读者可以一边看书，一边根据实训的要求，用编程软件和仿真软件进行操作。通过仿真实验，读者就能轻松掌握软件的操作方法和有关的知识点，并留下难忘的印象。

本书涵盖了 S7-300/400 应用技术主要的知识点，包括软件安装、硬件和网络组态、编程、监控、故障诊断、指令应用、程序结构、程序设计方法、通信、人机界面应用和 PID 控制等内容。做完全部实训后，读者就能较全面地掌握 S7-300/400 的使用方法。随书光盘中的 PID 例程可以用来学习整定 PID 参数的方法，实验结果用曲线显示，形象直观。

随书光盘提供了 STEP 7 V5.4 中文版、PLCSIM V5.4 SP3 和西门子人机界面的组态软件 WinCC flexible 2008，为仿真实验打下了基础。光盘中还有 S7-300/400 的用户手册。

建议一边阅读书中的实训，一边按实训中的叙述生成项目、组态硬件、编写程序和做仿真实验。如果已经熟悉了软件的操作方法，可以在了解例程的功能和读懂程序的基础上，直接运行随书光盘中比较复杂的例程，做仿真实验。

本书绝大多数实训都有仿真练习，读者可以在完成实训中的操作后，按仿真练习的要求做类似的或进一步的操作和练习，以巩固所学的知识。各章配有适量的习题。本书可以供工程技术人员自学，也可以作为高职高专、技工学校有关专业的教材。

本书由廖常初主编，范占华、关朝旺、余秋霞、陈曾汉、陈晓东、王云杰、李远树、廖亮、孙明渝、左源洁、万莉、郑群英、孙剑、唐世友参加了编写工作。

因作者水平有限，书中难免有错漏之处，恳请读者批评指正。

作者在中华工控网的博客网址为 http://blog.gkong.com/blog.asp?name=liaochangchu，欢迎读者访问和交流。

作者 E-mail：liaosun@cqu.edu.cn。

重庆大学 廖常初

目　　录

前言
第 1 章　概述 1
1.1　怎样学习 S7-300/400 PLC 1
1.2　实训一　安装 STEP 7 和仿真软件 PLCSIM 2
1.2.1　安装 STEP 7 2
1.2.2　安装 PLCSIM 6
第 2 章　STEP 7 使用入门 9
2.1　S7-300 系列 PLC 简介 9
2.1.1　S7-300 的系统结构 9
2.1.2　CPU 模块 10
2.1.3　信号模块 11
2.2　实训二　生成项目与组态硬件 14
2.2.1　生成一个新的项目 14
2.2.2　组态硬件 18
2.3　实训三　异步电动机正反转控制 23
2.3.1　生成用户程序 23
2.3.2　用 PLCSIM 调试程序 27
2.4　实训四　小车控制系统 31
2.4.1　PLC 的循环处理过程 31
2.4.2　小车控制系统 33
2.4.3　STEP 7 与 PLC 通信连接的组态 34
2.5　练习题 36
第 3 章　S7-300/400 的指令应用 38
3.1　位逻辑指令应用 38
3.1.1　实训五　位逻辑指令的仿真实验 38
3.1.2　实训六　故障显示电路 43
3.2　定时器计数器指令应用 45
3.2.1　实训七　定时器指令的基本功能 45
3.2.2　实训八　卫生间冲水控制电路 50
3.2.3　实训九　运输带控制系统 51
3.2.4　实训十　小车控制系统 54
3.2.5　实训十一　计数器指令的基本功能 55
3.3　数据处理指令与逻辑控制指令 58
3.3.1　实训十二　数据处理基础知识的仿真实验 58



3.3.2 实训十三 数据传送指令与语句表程序状态监控……61
3.3.3 实训十四 状态字的仿真实验……63
3.3.4 实训十五 逻辑控制指令的仿真实验……66
3.3.5 实训十六 比较指令的仿真实验……68
3.3.6 实训十七 数据转换指令的仿真实验……70
3.4 数学运算指令……73
3.4.1 实训十八 数学运算指令的仿真实验……73
3.4.2 实训十九 移位与循环移位指令的仿真实验……75
3.4.3 实训二十 字逻辑运算指令的仿真实验……77
3.5 练习题……78
第 4 章 S7-300/400 的用户程序结构……79
4.1 功能与功能块……79
4.1.1 S7-300/400 的用户程序结构……79
4.1.2 实训二十一 功能的生成与调用……80
4.1.3 实训二十二 功能块的生成与调用……82
4.1.4 实训二十三 共享数据块与系统功能的调用……86
4.1.5 实训二十四 多重背景的应用……89
4.2 组织块的应用……91
4.2.1 组织块与中断……91
4.2.2 实训二十五 使用循环中断的彩灯控制程序……93
4.2.3 实训二十六 时间中断组织块的应用……96
4.2.4 实训二十七 硬件中断组织块的应用……98
4.2.5 实训二十八 延时中断组织块的应用……101
4.3 练习题……103
第 5 章 梯形图的顺序控制设计法……104
5.1 顺序控制设计法……104
5.1.1 顺序功能图……104
5.1.2 实训二十九 使用置位复位指令的顺序控制程序……106
5.1.3 实训三十 专用钻床的顺序控制……108
5.2 S7-GRAPH 顺序功能图语言的应用……115
5.2.1 实训三十一 使用 S7-GRAPH 的运输带顺序控制……115
5.2.2 实训三十二 使用 S7-GRAPH 的剪板机顺序控制……122
5.3 实训三十三 生成与显示参考数据……126
5.4 练习题……130
第 6 章 网络通信……131
6.1 PROFIBUS-DP 网络通信……131
6.1.1 西门子工业通信网络简介……131
6.1.2 ET 200……132
6.1.3 实训三十四 组态分布式 I/O……134

6.1.4 实训三十五 组态 DP 主站与 S7-200 的通信……138
6.1.5 实训三十六 组态 DP 主站与智能从站的主从通信……141
6.1.6 实训三十七 用 SFC14 和 SFC15 传输一致性数据……145
6.2 S7 通信的组态与编程……148
6.2.1 S7 通信简介……148
6.2.2 实训三十八 基于 DP 网络的单向 S7 通信……149
6.2.3 实训三十九 基于 MPI 网络的双向 S7 通信……154
6.2.4 实训四十 上传项目文件……158
6.3 练习题……160
第 7 章 故障诊断……161
7.1 DP 从站与扩展机架的故障诊断……161
7.1.1 与 DP 通信有关的中断组织块……161
7.1.2 实训四十一 DP 从站的故障诊断……163
7.1.3 实训四十二 扩展机架的故障诊断……167
7.2 实训四十三 信号模块诊断功能的仿真实验……169
7.3 实训四十四 编程错误的诊断……175
7.4 练习题……178
第 8 章 人机界面的组态与仿真……179
8.1 人机界面的硬件与组态软件……179
8.1.1 人机界面与触摸屏……179
8.1.2 SIMATIC 人机界面……180
8.1.3 安装 WinCC flexible……181
8.2 实训四十五 人机界面的画面组态与仿真实验……182
8.2.1 创建项目和 HMI 站点……182
8.2.2 组态指示灯……185
8.2.3 组态按钮……188
8.2.4 组态文本域与 IO 域……190
8.2.5 PLC 与人机界面的集成仿真实验……191
8.3 练习题……193
第 9 章 PID 闭环控制……194
9.1 PID 闭环控制系统……194
9.1.1 模拟量闭环控制系统的组成……194
9.1.2 PID 控制器……195
9.1.3 过程变量的处理……196
9.1.4 PID 控制器的参数……196
9.1.5 控制器输出值的处理……198
9.2 实训四十六 PID 控制的编程与仿真实验……198
9.2.1 生成项目……198
9.2.2 PID 控制程序……202



9.2.3　PID 控制器的参数整定方法 ……206
9.2.4　PID 控制的仿真实验 ……209
9.3　练习题 ……212
附录　随书光盘内容简介 ……213
参考文献 ……215

第 1 章

概　述

1.1 怎样学习 S7-300/400 PLC

1．使用 S7-300/400 需要学什么

S7-300/400 是国内应用最广、市场占有率最高的大中型 PLC。使用 S7-300/400，需要掌握以下技能：

1）了解 S7-300/400 的硬件结构和网络通信功能。

2）熟练操作 S7-300/400 的编程软件 STEP 7，用它来完成对硬件和网络的组态、编程、调试和故障诊断等操作。

3）熟悉 S7-300/400 的指令系统和程序结构，能阅读和理解 PLC 的用户程序。

4）能编写、修改和调试用户程序。

2．学习 S7-300/400 的工具

S7-300/400 的硬件很贵，个人和一般的单位都很难有用大量的硬件来做实验的条件。

S7-PLCSIM 是 S7-300/400 功能强大、使用方便的仿真软件。可以用它在通用计算机上做实验，模拟 PLC 硬件的运行，包括执行用户程序。做仿真实验和做硬件实验时观察到的现象几乎完全相同。

本书的随书光盘提供了 STEP 7 V5.4 SP3.1 中文版、PLCSIM V5.4 SP3 和西门子人机界面的组态软件 WinCC flexible 2008，为仿真实验创造了条件。

3．学习 PLC 的主要方法是动手

如果不动手用编程软件和仿真软件（或 PLC 的硬件）进行操作，只是阅读教材或 PLC 的用户手册，不可能学会 PLC。

看十遍书不如动一次手，本书的特点是强调动手，强调实际操作。

本书的主体是四十多个实训，S7-300/400 应用的主要知识点都包含在这些实训里。通过仿真实验，读者能轻松地掌握编程软件和仿真软件的操作方法和有关的知识点，并且会留下难忘的印象。做完全部实训后，读者就能较全面地掌握 S7-300/400 的使用方法。

为了减少篇幅，本书尽量避免重复叙述相同的操作和出现相同的插图。如果读者是初学 S7-300/400，或者计算机基础较差，建议按顺序做书中的实训。

4．例程的使用方法

建议一边阅读书中的实训，一边按实训中的叙述生成项目、组态硬件、编写程序和做仿真实验。随书光盘有三十多个与正文配套的例程，如果已经熟悉了软件的操作方法，可以在了解例程的功能和读懂程序的基础上，直接运行比较复杂的例程，做仿真实验。

5．在线帮助功能的使用

STEP 7 有非常强大的在线帮助功能，打开某个对话框的某个选项卡、选中菜单中的某条命令、或选中指令列表或程序中的某条指令或程序块，按计算机的〈F1〉键，就能得到有关对象的在线帮助信息。建议读者充分利用在线帮助信息来解决遇到的问题。

如果读者有较强的计算机基础知识，对 S7-300/400 的应用已经有一定的基础，在学习本书的同时，可以阅读作者编写的《S7-300/400 PLC 应用技术》。该书是第一本全面、系统地介绍 S7-300/400 的书籍，获 2006 年度机械工业出版社科技进步奖，在华储网上电脑书店居自动化类图书销售排行榜第一名。

1.2 实训一 安装 STEP 7 和仿真软件 PLCSIM

1.2.1 安装 STEP 7

1．对计算机的要求

安装 STEP 7 V5.4 SP3.1 中文版对计算机的要求如下：操作系统必须是 Windows XP Professional（专业版），不能在 Windows XP Home（家用）版安装。CPU 的主频在 600MHz 以上，内存大于等于 512MB，推荐 1GB。显示器支持 1024×768 的分辨率，16 位彩色。

建议操作系统采用深度 Windows XP SP3，它对西门子各主要软件的支持较好，采用某些版本附带的 Ghost 功能可将 C 盘压缩为*.Gho 文件后，保存在别的磁盘分区。操作系统或安装在 C 盘的软件有问题时，可以用 Ghost 恢复备份的 C 盘。

建议将 STEP 7 和西门子的其他大型软件（例如 WinCC flexible 和 WinCC 等）安装在 C 盘。这些软件出现问题时，可以用 Ghost 快速恢复它们。西门子的软件一般有 14 天的试用期，安装后用 Ghost 备份 C 盘，试用期结束时用 Ghost 恢复 C 盘，又可以获得 14 天的试用期。

如果 C 盘安装的大型软件较多，占用的空间较大，在将硬盘分区时，建议给 C 盘分配大于 10GB 的空间。

2．安装 STEP 7

打开随书光盘的文件夹“\STEP7_V54_SP3_CH\CD_1”，双击其中的安装文件 Setup.exe（其图标为），开始安装。结束每个对话框的操作后，单击“下一步 >”按钮，打开下一个对话框。有的对话框没有什么操作，只需要单击“下一步 >”按钮确认。

在第一页确认安装使用的语言为默认的简体中文（见图 1-1 的左图）。

设置要安装的软件时，默认的选择是安装图 1-1 右边的 3 个软件，必须安装 STEP 7 和 Automation License Manager（自动化许可证管理器）。建议采用默认的安装路径（C:\Program Files \Siemens）。

Adobe Reader 8 用于阅读用户手册，建议不要安装它（用鼠标单击它左边的复选框，使其中的“√”消失），因为在互联网上可以搜索和下载它的更高的版本。安装某些操作系统时，可能会自动安装阅读用户手册的软件。

单击“下一步”按钮，出现图 1-1 中的小对话框，显示出需要安装的软件，正在安装的软件用加粗的字体表示，此前已经安装好的软件不会在该对话框中出现。“.Net Framework”



是微软公司的一个软件。

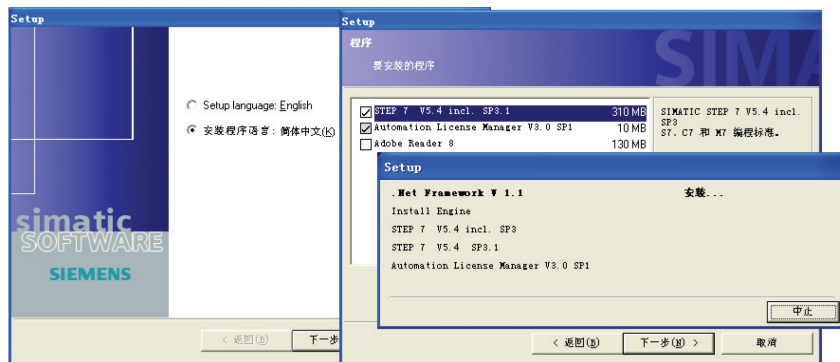


图 1-1　设置安装语言与需要安装的软件

在安装 STEP 7 之前，首先出现图 1-2 左边的欢迎画面，单击“说明文件”对话框中的“我要阅读注意事项”按钮（见图 1-2 的右图），将打开软件的说明文件。

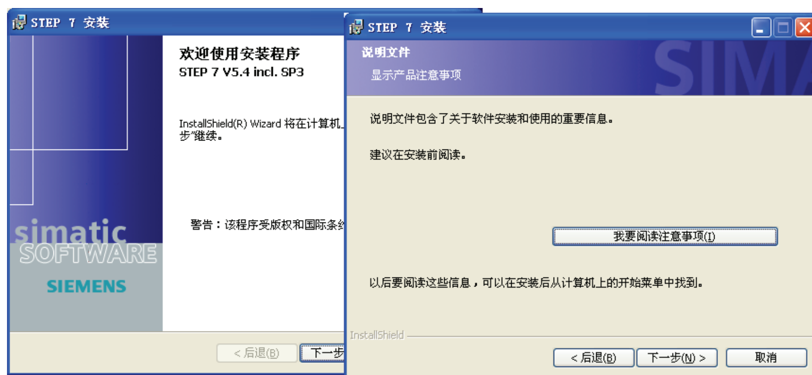


图 1-2　欢迎和说明文件对话框

在“许可协议”对话框（见图 1-3 的左图），应选中“接受许可协议的条款”。

在“用户信息”对话框（见图 1-3 的右图），可以输入用户信息，或采用默认的设置。

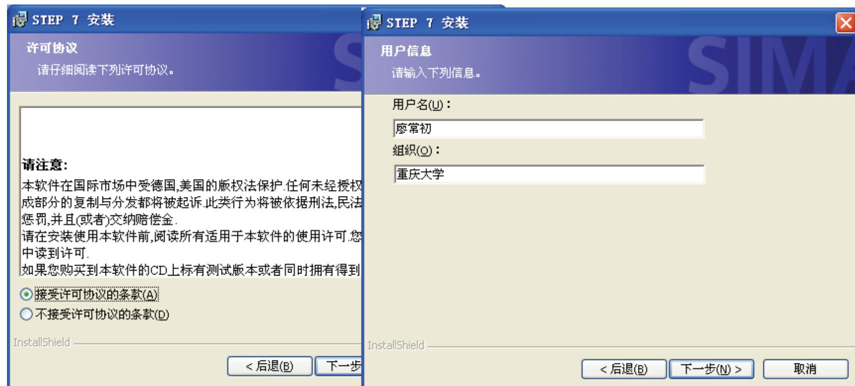


图 1-3　许可协议与用户信息对话框

在“安装类型”对话框（见图 1-4 的左图），建议采用默认的安装类型（典型的）和默认的安装路径。单击“更改”按钮，可以改变安装 STEP 7 的文件夹。修改后单击“确定”按钮，返回“安装类型”对话框。

在“产品语言”对话框（见图 1-4 的右图），英语是默认的语言，此外选中“简体中文”复选框，因此将安装两种语言。

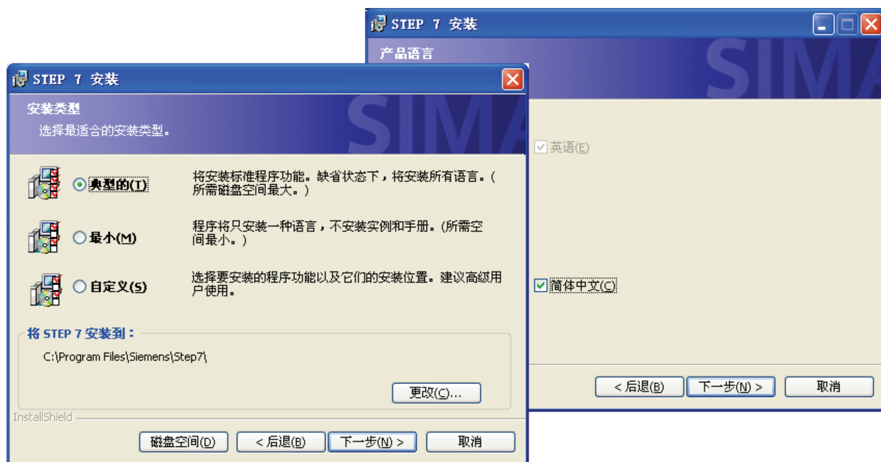


图 1-4　安装类型与产品语言对话框

在“传送许可证密钥”对话框（见图 1-5 的左图），选中“否，以后再传送许可证密钥”，以后可以用许可证管理器安装许可证密钥。如果没有许可证密钥，可以在首次打开安装好的软件时，激活 14 天期限的试用许可证密钥（见图 2-13）。

单击图 1-5 右图的“安装”按钮，开始安装 STEP 7。

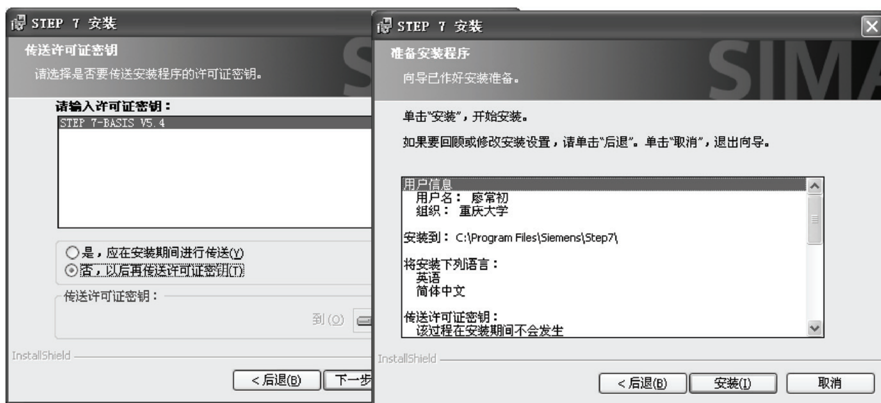


图 1-5　传送许可证密钥与准备安装程序对话框

安装好软件后，出现“安装/删除接口”对话框（见图 1-6 的左图），可以安装或卸载编程计算机与 PLC 通信的硬件的驱动程序。

在“设置 PG/PC 接口”对话框（见图 1-6 的右图），可以设置选中的计算机接口的参数。安装好 STEP 7 后，执行 SIMATIC 管理器的菜单命令“选项”→“设置 PG/PC 接口”，也可以



打开该对话框。单击该对话框中的“选择”按钮，可以打开“安装/删除接口”对话框。如果只做仿真实验，单击图 1-6 中的“关闭”和“确定”按钮，关闭这两个对话框。

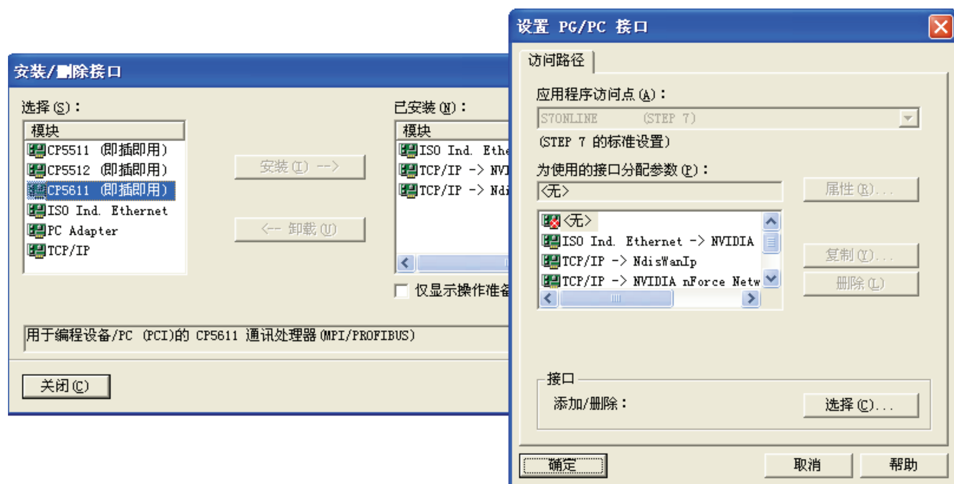


图 1-6　设置通信接口的对话框

出现与防火墙有关的两个对话框时（见图 1-7），单击“是”按钮确认。

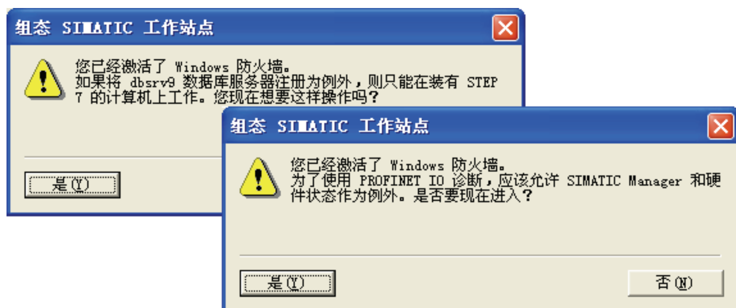


图 1-7　与防火墙有关的对话框

安装完成后，在出现的对话框中（见图 1-8），采用默认的选项“是，立即重启计算机”。单击“完成”按钮结束安装过程。

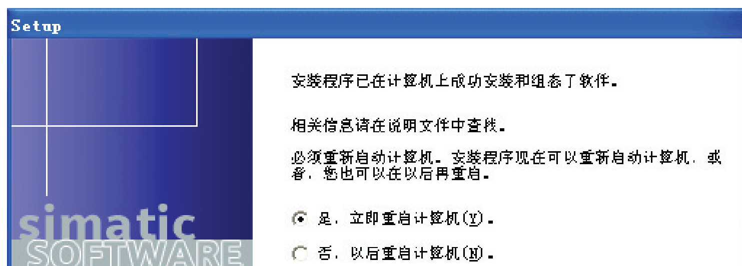


图 1-8　安装完成对话框

3. 安装 STEP 7 的注意事项

1）可以用随书光盘安装 STEP 7 和 PLCSIM，也可以将软件复制到硬盘后再安装，但是

保存它们的文件夹的层次不能太多，各级文件夹的名称不能使用中文，否则在安装时可能会出现“ssf 文件错误”的信息。

2）在安装 STEP 7 或其他软件时，可能出现“Please restart Windows before installing new programs”（安装新程序之前，请重新启动 Windows），或其他类似的信息。如果重新启动计算机后再安装软件，还是出现上述信息，说明因为杀毒软件的作用，Windows 操作系统已经注册了一个或多个写保护文件，以防止被删除或重命名。解决的方法如下：

执行 Windows 的菜单命令“开始”→“运行”，在出现的“运行”对话框中输入“regedit”，打开注册表编辑器。选中注册表左边的“HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager”，如果右边窗口中有条目“PendingFile Rename Operations”，将它删除，不用重新启动计算机，就可以安装软件了。可能每安装一个软件都需要做同样的操作。

3）注意西门子自动化软件的安装顺序。必须先安装 STEP 7，再安装上位机组态软件 WinCC 和人机界面的组态软件 WinCC flexible。

1.2.2 安装 PLCSIM

1. 安装 PLCSIM V5.4

双击随书光盘的“\PLCSIM_V54”文件夹中的文件 Setup.exe，开始安装仿真软件 PLCSIM。在安装语言对话框（见图 1-9 的左图），采用默认的设置，安装语言为英语。完成各对话框中的设置后，单击“Next”（下一步）按钮确认。

因为在安装 STEP 7 时已经安装了 Automation License Manager（自动化许可证管理器），采用图 1-9 的右图（需安装的程序对话框）中默认的设置，只安装 PLCSIM。

单击“Browse”（浏览）按钮，可以修改安装 PLCSIM 的文件夹。建议采用默认的目标文件夹。

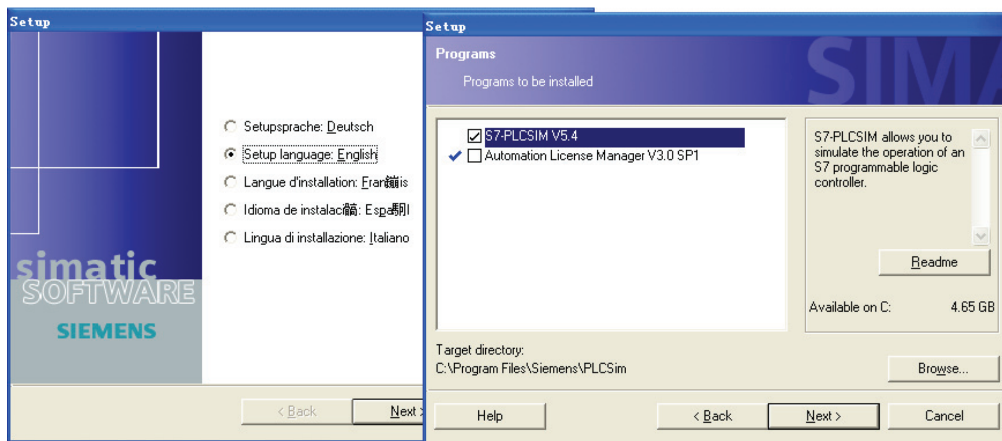


图 1-9 设置安装语言与需要安装的软件

图 1-10 的左图是欢迎画面，单击图 1-10 的右图（说明文件对话框）中间的长按钮，将打开说明文件。

在“许可协议”对话框（见图 1-11 的左图），应选中“I accept…”（接受许可协议的条款）。



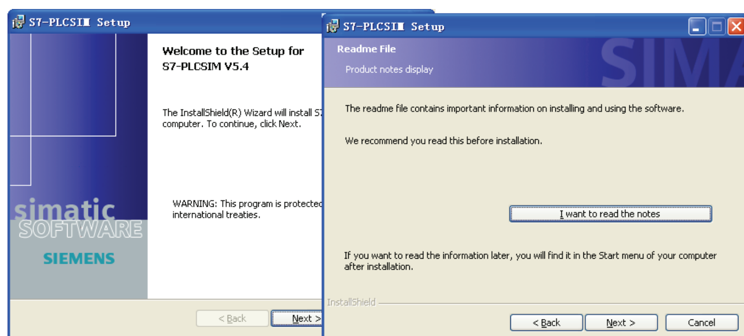


图 1-10 欢迎与说明文件对话框

在“安装类型”对话框（见图 1-11 的右图），建议采用默认的选项“Typical”（典型安装）和默认的目标文件夹。

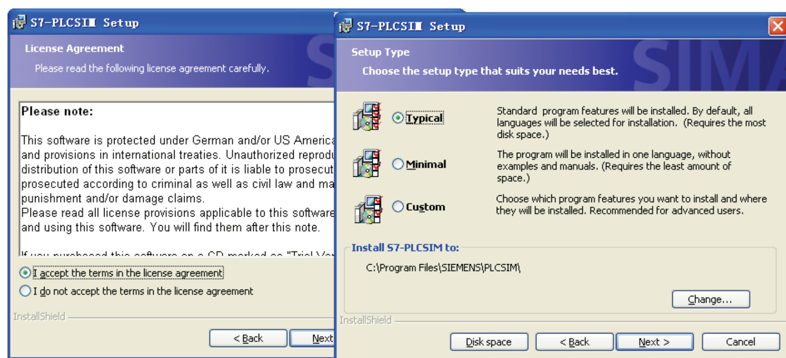


图 1-11 许可协议与安装类型对话框

在产品语言对话框，默认的选项是安装 5 种欧洲语言，用复选框取消英语之外的其他语言（见图 1-12 的左图）。

在“Transfer License Keys”（传送许可证密钥）对话框（见图 1-12 的右图），选中“No, Transfer License Keys later”（不，以后再传送许可证密钥）。

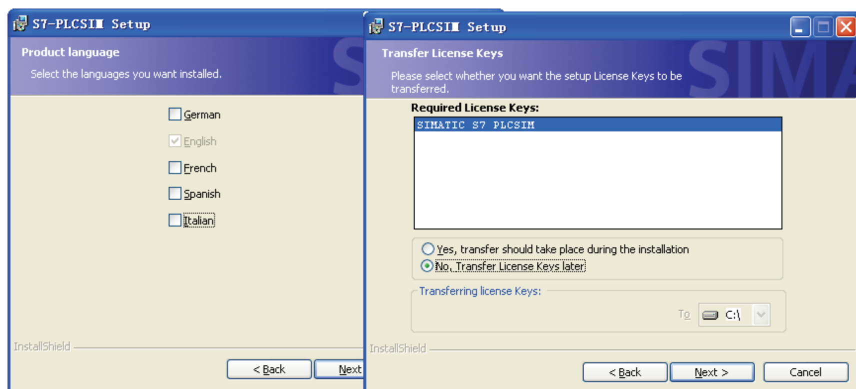


图 1-12 产品语言与传送许可证密钥对话框

单击图 1-13 左图中的“Install”（安装）按钮，开始安装 PLCSIM。

单击最后出现的安装完成对话框中的“Finish”（结束）按钮（见图 1-13 的右图），结束安装过程。

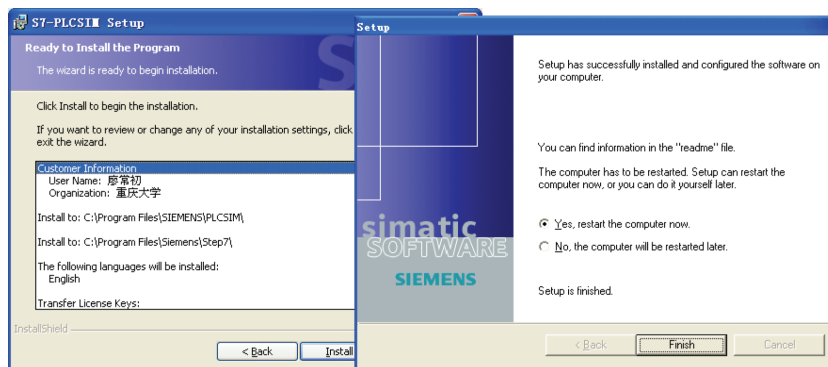


图 1-13　准备好安装软件与安装完成对话框

2．安装 PLCSIM V5.4 SP3

双击随书光盘的“\PLCSIM V54 SP3”文件夹中的文件 Setup.exe，开始安装 PLCSIM V5.4 的升级包 SP3。安装过程与安装 PLCSIM 基本相同。

在安装语言对话框（见图 1-9 的左图），采用默认的设置，安装语言为英语。在“Programs”对话框，确认自动选中的需要安装的软件。

在“许可协议”对话框（见图 1-11 的左图），应选中“I accept…”（接受许可协议的条款）。

在“安装类型”对话框（见图 1-14 的左图），采用默认的选项“Upgrade”（升级）。

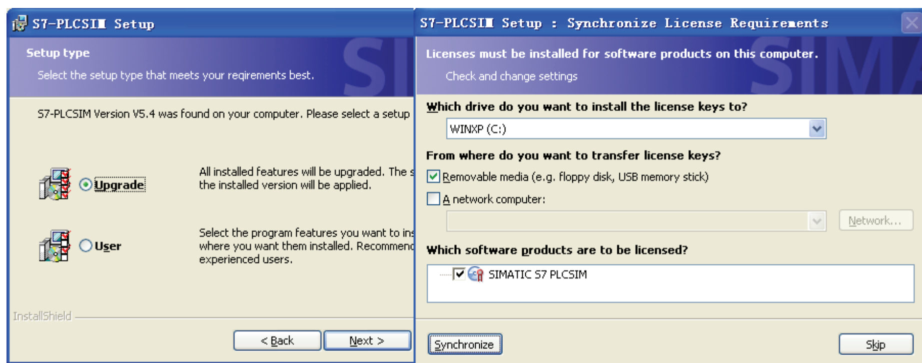


图 1-14　安装类型与安装许可证密钥的对话框

单击图 1-13 左图中的“Install”（安装）按钮，开始安装 PLCSIM。

单击图 1-14 右图中的“Skip”按钮，不安装软件的许可证密钥。

单击最后出现的安装完成对话框中的“Finish”（结束）按钮（见图 1-13 的右图），结束安装过程。

第 2 章

STEP 7 使用入门

2.1　S7-300 系列 PLC 简介

2.1.1　S7-300 的系统结构

S7-300 是模块化的中小型 PLC, 适用于中等性能的控制要求。

S7-300 的 CPU 都有一个使用 MPI（多点接口）通信协议的 RS-485 接口。有的 CPU 还带有集成的现场总线 PROFIBUS-DP 接口、PROFINET 接口或 PtP（点对点）串行通信接口。

功能最强的 CPU 319-3PN/DP 的 RAM 存储容量为 1400KB，可以插入 8M 的微存储卡（MMC），有 8192B 存储器位，2048 个 S7 定时器和 2048 个 S7 计数器，数字量输入和输出最多均为 65536 点，模拟量输入和输出最多均为 4096 点，位操作指令的执行时间为 0.01μs。

S7-300/400 有很高的电磁兼容性和抗振动抗冲击能力。可以用于恶劣环境的 SIPLUS S7-300 的使用温度范围为 –25～+70°C，有更强的耐振动和耐污染性能。

S7-300/400 有 350 多条指令，其编程软件 STEP 7 功能强大，可以使用多种编程语言。STEP 7 可为所有的模块和网络设置参数。

CPU 用智能化的诊断系统连续监控系统的功能是否正常、记录错误和特殊系统事件(例如超时、模块更换等)。S7-300 有过程报警和多种中断功能。

S7-300 采用紧凑的、无槽位限制的模块结构（见图 2-1），电源模块（PS）、CPU 模块、信号模块（SM）、功能模块（FM）、接口模块（IM）和通信处理器（CP）都安装在铝制导轨上。图 2-2 是 S7-400 机架上的电源模块和 CPU 模块。



图 2-1　S7-300



图 2-2　S7-400 的电源模块和 CPU 模块

S7-300 的电源模块总是安装在机架的最左边，CPU 模块紧靠电源模块。

S7-300 用背板总线将除电源模块之外的各个模块连接起来。背板总线集成在模块上，模块通过 U 形总线连接器相连，每个模块（不包括电源模块）都有一个总线连接器，后者插在各模块的背后（见图 2-3）。安装时先将总线连接器插在 CPU 模块上，将模块固定在导轨上，然后依次安装各个模块。

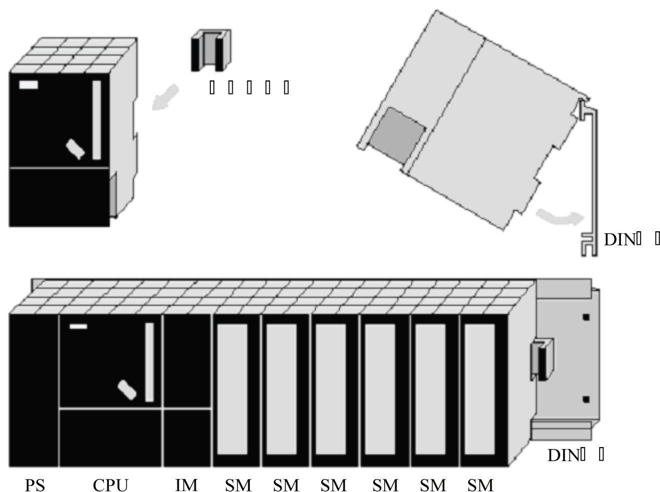


图 2-3　S7-300 的安装

除了带 CPU 的中央机架，最多可以增加 3 个扩展机架，每个机架最多可以插 8 块信号模块、功能模块或通信处理器模块。组态时系统自动分配模块的地址。

电源模块在机架最左边的 1 号槽（见图 2-3）。中央机架（0 号机架）的 2 号槽是 CPU 模块，3 号槽是接口模块。其他模块使用 4～11 号槽。

机架导轨上并不存在物理槽位，在不需要扩展机架时，CPU 模块和 4 号槽的模块是挨在一起的。此时 3 号槽位仍然被实际上并不存在的接口模块占用。

PS 307 电源模块将 AC 120/230V 电压转换为 DC 24V 电压，为 S7-300、传感器和执行器供电。额定输出电流有 2A、5A 和 10A。

电源模块除了给 CPU 模块提供电源外，还要给输入/输出模块提供 DC 24V 电源。

2.1.2　CPU 模块

S7-300 有多种不同型号的 CPU，分别适用于不同等级的控制要求。有的 CPU 模块集成了数字量 I/O，有的同时集成了数字量 I/O 和模拟量 I/O。

CPU 内的元件封装在一个牢固而紧凑的塑料机壳内，面板上有状态和错误指示 LED、模式选择开关和通信接口（见图 2-4）。微存储卡插槽可以插入多达数兆字节的 FEPROM 微存储卡(MMC)，用于掉电后用户程序和数据的保存。有的 CPU 只有一个 MPI 接口。

1．状态与故障显示 LED

CPU 模块面板上的 LED（发光二极管）的意义见表 2-1。



表 2-1　S7-300 CPU 的 LED 指示灯

指示灯	颜色	说明	LED 指示灯	颜色	说明
SF	红色	系统错误/故障	FRCE	黄色	有输入/输出处于被强制的状态
BF	红色	通信接口的总线故障	RUN	绿色	CPU 处于运行模式
DC 5V	绿色	5V 电源正常	STOP	黄色	CPU 处于停止模式

2．CPU 的操作模式

1）STOP（停机）模式：模式选择开关在 STOP 位置时，CPU 模块上电后自动进入 STOP 模式，在该模式不执行用户程序，可以接收全局数据和检查系统。

2）RUN（运行）模式：执行用户程序，刷新输入和输出，处理中断和故障信息服务。

3）HOLD 模式：在起动和 RUN 模式执行程序时遇到调试用的断点，用户程序的执行被挂起（暂停），定时器被冻结。

4）STARTUP（启动）模式：可以用模式选择开关或 STEP 7 起动 CPU。如果模式选择开关在 RUN 位置，通电时自动进入启动模式。

5）老式的 CPU 还有一种 RUN-P 模式，允许在运行时读出和修改程序。仿真软件 PLCSIM 的仿真 CPU 也有 RUN-P 模式，某些监控功能只能在 RUN-P 模式进行。

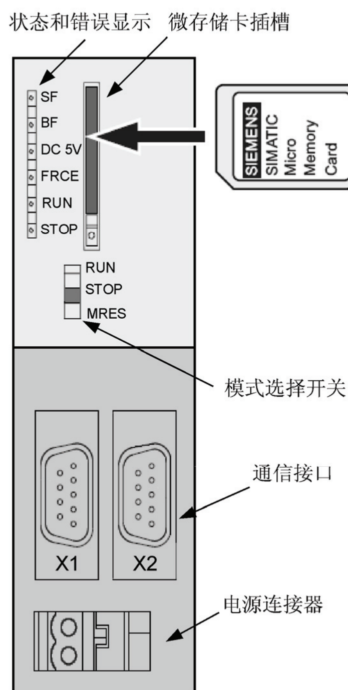


图 2-4　CPU 315-2DP 的面板

2.1.3　信号模块

信号模块（SM）包括数字量（或称开关量）输入（DI）模块、数字量输出（DO）模块、数字量输入/输出（DI/DO）模块、模拟量输入（AI）模块、模拟量输出（AO）模块和模拟量输入/输出（AI/AO）模块。

信号模块和功能模块的外部接线接在插接式的前连接器的端子上，前连接器插在前盖板后面的凹槽内。

模块面板上的 SF LED 用于显示故障和错误，数字量 I/O 模块面板上的 LED 用来显示各数字量输入/输出点的信号状态。模块安装在标准导轨上，并通过总线连接器与相邻模块连接（见图 2-3）。

1．数字量输入模块

数字量输入模块用于连接外部的机械触点和电子数字式传感器，例如光电开关和接近开关，将来自现场的外部数字量信号的电平转换为 PLC 内部的信号电平。

图 2-5 是直流输入模块的内部电路和外部接线图，图中只画出了一路输入电路，M 或 N 是同一输入组各内部输入电路的公共点。

当图 2-5 中的外接触点接通时，光耦合器中的发光二极管点亮，光敏三极管饱和导通；外接触点断开时，光耦合器中的发光二极管熄灭，光敏三极管截止，信号经背板总线接口传

送给 CPU 模块。

交流输入模块的额定输入电压为 AC 120V 或 230V。图 2-6 的电路用电阻、电容降压，经桥式整流电路转换为直流电流。外接触点接通时，光耦合器中的发光二极管和显示用的发光二极管点亮，光敏三极管饱和导通。外接触点断开时，光耦合器中的发光二极管熄灭，光敏三极管截止，相当于触点断开。

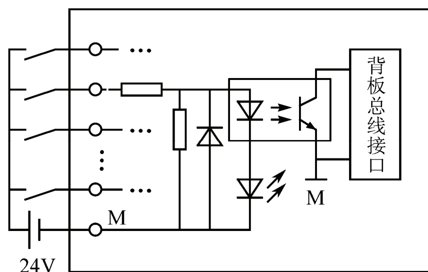


图 2-5　数字量直流输入模块电路

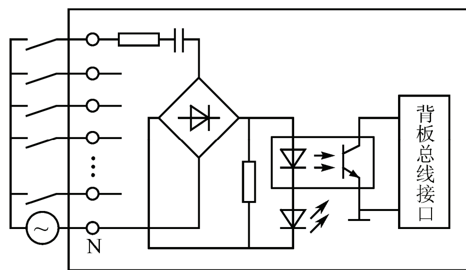


图 2-6　数字量交流输入模块电路

直流输入电路的延迟时间较短，可以直接与接近开关、光电开关等电子传感器连接。交流输入方式适合于在有油雾、粉尘的恶劣环境下使用。

2．数字量输出模块

数字量输出模块用于驱动电磁阀、接触器、小功率电动机、指示灯和电动机起动器等负载。数字量输出模块将内部信号电平转化为控制过程所需的外部信号电平，同时有隔离和功率放大的作用。负载电源由外部现场提供。

图 2-7 是继电器输出电路，某一输出点为 1 状态时，梯形图中的线圈“通电”，通过背板总线接口和光耦合器，使模块中对应的微型继电器线圈通电，其常开触点闭合，使外部负载工作。输出点为 0 状态时，梯形图中的线圈“断电”，输出模块中对应的微型继电器的线圈也断电，其常开触点断开。继电器输出模块既可以驱动交流负载，也可以驱动直流负载。

固态继电器（SSR）输出模块（图 2-8）只能驱动交流负载，小方框内的光敏双向晶闸管和小方框外的双向晶闸管等组成固态继电器。SSR 的输入功耗低，输入信号电平与 CPU 内部的电平相同，同时又实现了隔离，并且有一定的带负载能力。梯形图中某一输出点为 1 状态时，其线圈“通电”，使光敏晶闸管中的发光二极管点亮，光敏双向晶闸管导通，使另一个容量较大的双向晶闸管导通，模块外部的负载得电工作。图 2-8 中的 RC 电路用来抑制晶闸管的关断过电压和外部的浪涌电压。这类模块的开关速度较快，工作寿命长。

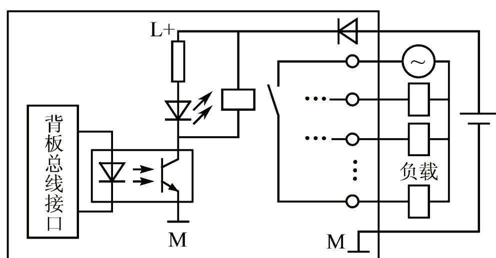


图 2-7　继电器输出模块电路

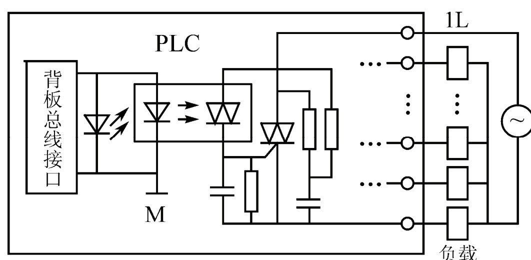


图 2-8　固态继电器输出模块电路



图 2-9 是晶体管与场效应晶体管输出电路，只能驱动直流负载。输出信号经光耦合器送给输出元件，图中用一个带三角形符号的小方框表示输出元件。输出元件的饱和导通状态和截止状态相当于触点的接通和断开。

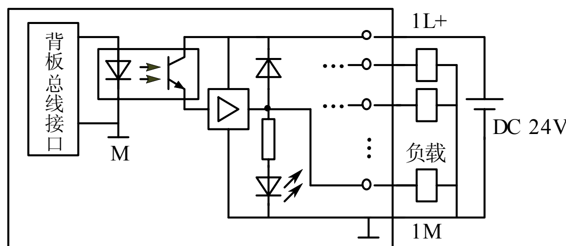


图 2-9　晶体管与场效应晶体管输出模块电路

继电器输出模块的负载电压范围宽，导通压降小，　受　时过电压和　时过电流的能力较强，但是动作速度较慢，寿命(动作次数)有一定的限制。如果系统输出量的变化不是很频繁，建议优先选用继电器型的输出模块。

固态继电器型输出模块只能用于交流负载，晶体管型、场效应晶体管型输出模块只能用于直流负载，它们的可靠性高，响应速度快，寿命长，但是过载能力稍差。

3．模拟量输入模块

生产过程中有大量的连续变化的模拟量需要用 PLC 来测量或控制。有的是非电量，例如温度、压力、流量、液位、物体的成分和频率等。有的是强电电量，例如发电机组的电流、电压、有功功率和无功功率等。变送器用于将传感器提供的电量或非电量转换为标准量程的直流电流或直流电压信号，例如 DC 0～10V 和 DC 4～20mA。

模拟量输入（AI）模块用于将模拟量信号转换为 CPU 内部处理用的数字信号，其主要组成部分是 A/D (Analog/Digit)转换器（见图 2-10）。AI 模块的输入信号一般是变送器输出的标准量程的直流电压、直流电流信号，有的也可以直接连接不带附加放大器的温度传感器（热电偶或热电阻），这样可以省去温度变送器。

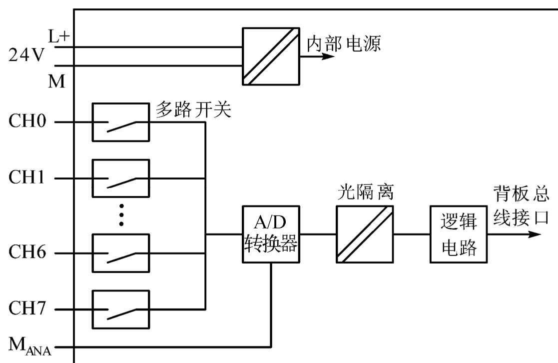


图 2-10　模拟量输入模块示意图

AI 模块中的各个通道可以分组设置为电流输入或电压输入，并选用不同的量程。

AI 模块由多路开关、A/D 转换器（ADC）、光隔离元件、内部电源和逻辑电路组成（见图 2-10）。各模拟量输入通道共用一个 A/D 转换器，用多路开关切换被转换的通道，AI 模块各输入通道的 A/D 转换过程和转换结果的存储与传送是顺序进行的。各个通道的转换结果被保存到各自的存储器，直到被下一次的转换值覆盖。

模块对热电偶、热电阻输入信号进行了线性化处理。使用屏蔽电缆时最大距离为 200m，输入信号为 50mV 或 80mV 时，最大距离为 50m。

AI 模块用量程卡（或称为量程模块）来切换不同类型的输入信号的输入电路。量程卡安装在模拟量输入模块的侧面，每两个通道为一组，共用一个量程卡（见图 2-11）。量程卡插入输入模块后，如果量程卡上的标记 C 与 AI 模块上的箭头相对应，则量程卡被设置在 C 位置。

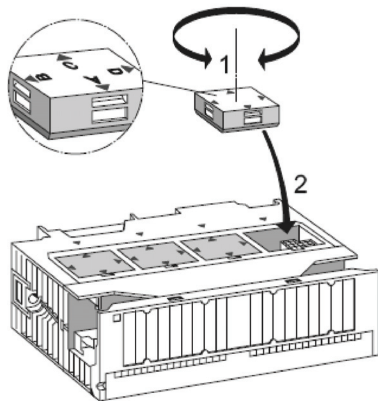


图 2-11　量程卡

对模块组态时，可以获得所选量程的量程卡的位置。设置量程卡时先用螺钉旋具将量程卡从模拟量输入模块中撬出来，再按组态的要求将量程卡插入模拟量输入模块。

4．模拟量输出模块

S7-300 的模拟量输出（AO）模块用于将 CPU 传送给它的数字转换为成比例的电流信号或电压信号，对执行机构进行调节或控制，其主要组成部分是 D/A 转换器（图 2-12 中的 DAC）。

AO 模块均有诊断中断功能，用红色 LED 指示故障。模块与背板总线有光隔离，使用屏蔽电缆时最大距离为 200m。

AO 模块为负载和执行器提供电流和电压，模拟信号应使用屏蔽电缆或双绞线电缆来传送。电缆线 QV 和 S_+、M_{ANA} 和 S_-（见图 2-12）应分别绞接在一起，这样可以减轻干扰的影响，另外应将电缆两端的屏蔽层接地。

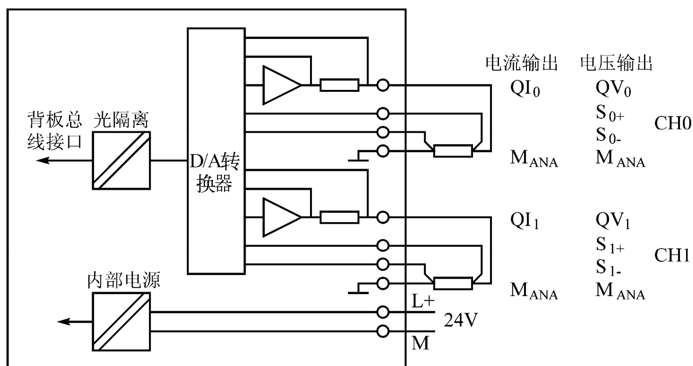


图 2-12　模拟量输出模块电路

2.2　实训二　生成项目与组态硬件

2.2.1　生成一个新的项目

1．激活试用许可证密钥

双击计算机桌面上的 STEP 7 图标，打开 SIMATIC Manager（SIMATIC 管理器）。第一次打开 STEP 7 时，出现图 2-13 所示的对话框，选中“STEP 7-Basis”，“激活”按钮上字符的颜色变为黑色，单击它将激活期限为 14 天的试用许可证密钥。



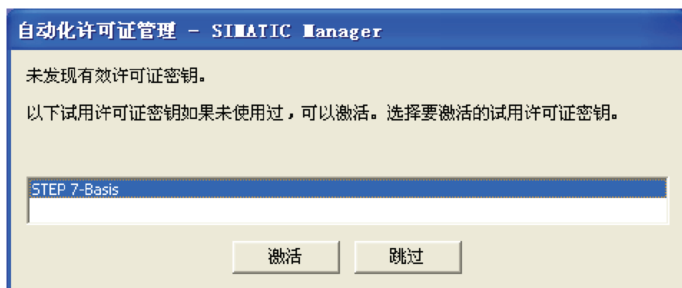


图 2-13　激活试用许可证密钥

双击计算机桌面上的图标，打开自动化许可证管理器（见图 2-14）。单击左边窗口的 C 盘，在右边窗口可以看到 C 盘上的试用（Trial）许可证，括号中的 14 天是允许使用的天数，括号前面是已经使用的天数。上述操作不是使用软件必须的操作。



图 2-14　自动化许可证管理器

2. 用新建项目向导创建项目

用鼠标左键单击（简称为“单击”）SIMATIC 管理器的“文件”菜单（见图 2-15），向下移动鼠标，选中“‘新建项目’向导”命令，其背景变为深蓝色。

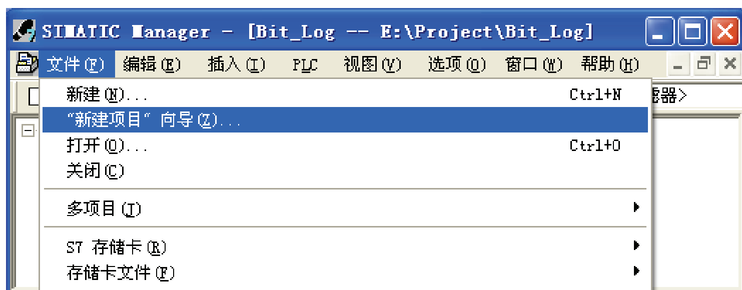


图 2-15　执行新建项目向导菜单命令

单击该命令，打开“STEP 7 向导：‘新建项目’”对话框（见图 2-16 的左图），菜单自动关闭。对话框中的小方框（称为复选框）中有“√”，表示选中了“在启动 SIMATIC Manager 时显示向导”。以后在打开 STEP 7 时，都将自动打开新建项目向导。单击向导中的“取消”按钮，将会打开上次关闭 STEP 7 时打开的项目。

单击“下一个>”按钮，在下一对话框中选择 CPU 模块的型号（见图 2-16 的右图）。组态实际的系统时，CPU 的型号与订货号应与实际的硬件相同，CPU 列表框的下面是所选 CPU 的基本特性。

单击“预览”按钮，可以打开或关闭该按钮下面的项目预览窗口（见图 2-16 的右图）。



图 2-16 新建项目向导

单击“下一个>”按钮，在下一对话框中选择需要生成的组织块（OB，见图 2-17 的左图）。一般采用默认的设置，只生成主程序 OB1。



图 2-17 新建项目向导

默认的编程语言为语句表（STL），用单选框将它修改为梯形图（LAD）。单选框的每个选项左边有一个小圆圈，选中某个选项时，小圆圈内出现小圆点。同时只能选中单选框的一个选项，但是可以同时选中多个复选框。

单击“下一个>”按钮，可以在“项目名称”文本框修改默认的项目名称（见图 2-17 的右图）。单击“完成”按钮，开始创建项目。



项目的名称最多允许 8 个字符，每个中文占两个字符。读者可以“故意”设置项目的名称超过 8 个字符，保存项目后，观察保存项目的文件夹的名称的长度。

3．项目的分层结构

项目是以分层结构保存对象数据的文件夹，包含了自动控制系统中所有的数据，图 2-18 的左边是项目树形结构窗口。第一层为项目，第二层为站，站是组态硬件的起点。站的下面是 CPU，“S7 程序”文件夹是编写程序的起点，所有的用户程序均存放在该文件夹中。

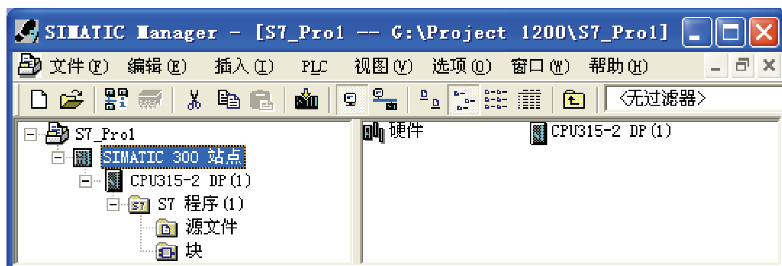


图 2-18　SIMATIC 管理器

用鼠标选中项目结构中某一层的对象，管理器右边的窗口将显示所选文件夹内的对象。双击其中的某个对象，可以打开和编辑该对象。

项目包含站和网络对象，站包含硬件、CPU 和 CP（通信处理器），CPU 包含 S7 程序和连接，S7 程序包含源文件、块和符号表。生成程序时自动生成一个空的符号表。

项目刚生成时，“块”文件夹中只有主程序 OB1。

4．设置项目属性

STEP 7 中文版可以使用中文和英语（默认的是中文）。可以用下面的方法修改为英语：执行 SIMATIC 管理器中的菜单命令“选项”→“自定义”，选中出现的“自定义”对话框的“语言”选项卡中（见图 2-19 的左图）的“english”。单击“确定”按钮，将自动退出 STEP 7。重新打开它后，项目语言（包括帮助文件）变为英语。

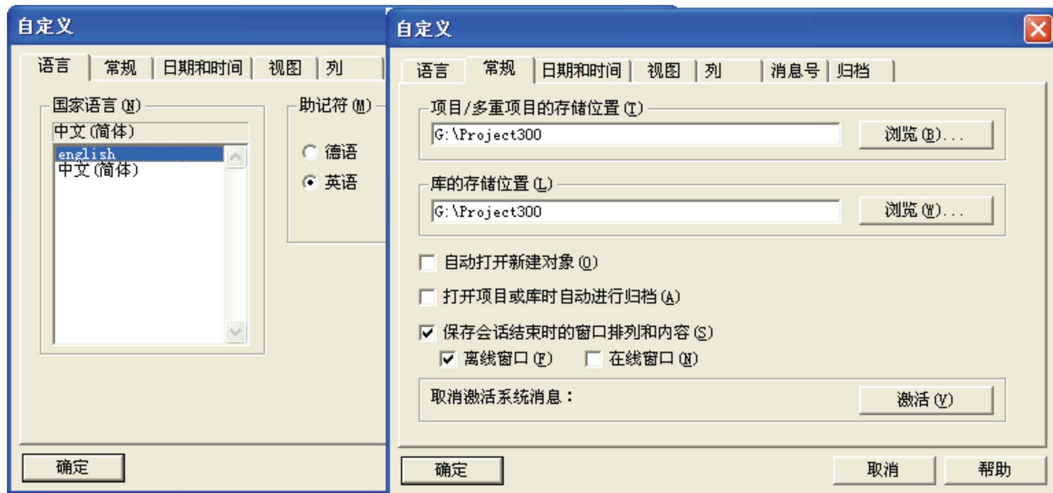


图 2-19　自定义对话框

可以用该选项卡的单选框选择使用德语或英语的助记符。

在“常规”选项卡（见图 2-19 的右图），可以修改保存项目和库的文件夹。如果保存项目的文件夹的名称中有中文，不能使用“新建项目”向导。

建议项目不要保存在 C 盘，因为用 Ghost 恢复 C 盘时，将会丢失 C 盘中原有的所有文件。

2.2.2 组态硬件

1. 硬件组态工具 HW Config

硬件组态的任务就是在 STEP 7 中生成一个与实际的硬件系统完全相同的系统，例如生成网络和网络中的各个站；生成 PLC 的机架，在机架中插入模块，以及设置各站点或模块的参数，即给参数赋值。

硬件组态确定了 PLC 输入/输出变量的地址，为设计用户程序打下了基础。

选中 SIMATIC 管理器左边的站对象，双击右边窗口的“硬件”图标（见图 2-18），打开硬件组态工具 HW Config（见图 2-20）。

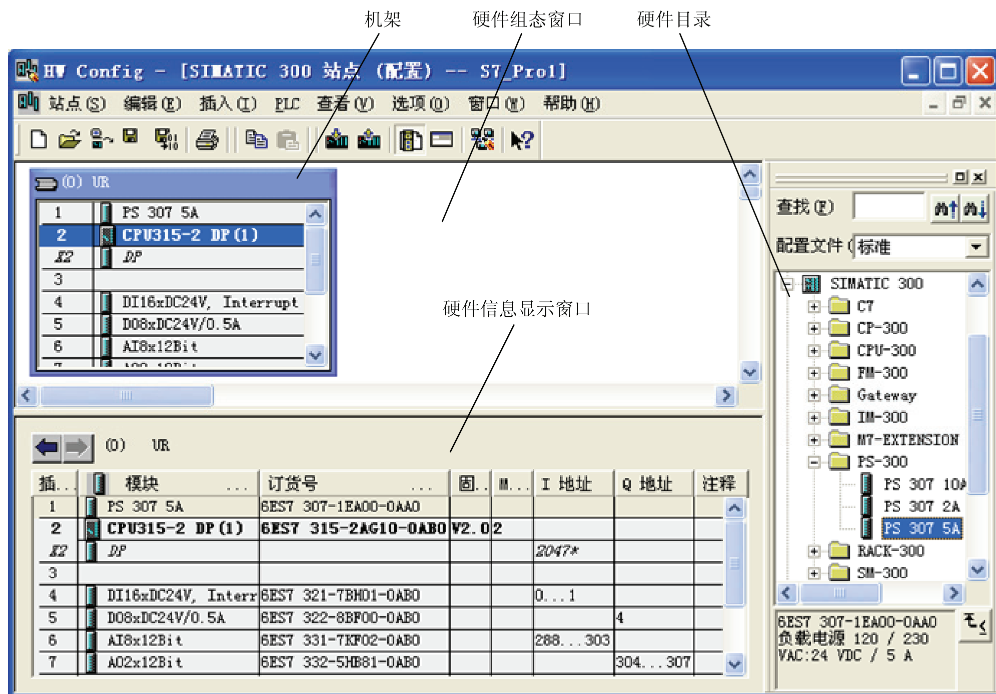


图 2-20 硬件组态窗口

刚打开 HW Config 时，左上方的硬件组态窗口中只有“新建项目”向导自动生成的机架，和 2 号槽中的 CPU 模块。右边是硬件目录窗口，可以用工具栏上的目录按钮打开或关闭它。选中硬件目录中的某个硬件对象，硬件目录下面的小窗口是它的订货号和简要的信息。

S7-300 的电源模块必须放在 1 号槽，2 号槽是 CPU 模块，3 号槽是接口模块，4～11 号槽放置其他模块。如果只有一个机架，3 号槽空着，但是实际的 CPU 模块和 4 号槽的模块紧挨着。

单击项目窗口中“SIMATIC 300”文件夹左边的⊞，打开该文件夹，其中的 CP 是通信处理器，FM 是功能模块，IM 是接口模块，PS 是电源模块，RACK 是机架，SM 是信号模块。单击某文件夹左边的⊟，将关闭该文件夹。



2．放置硬件对象的方法

组态时用组态表来表示机架或导轨，可以用鼠标将右边硬件目录窗口中的模块放置到组态表的某一行，就好像将真正的模块插入机架的某个槽位一样。

（1）用“拖放”的方法放置硬件对象

用鼠标打开硬件目录中的文件夹“\SIMATIC 300\PS-300”，单击其中的电源模块“PS 307 5A”，该模块被选中，其背景变为深色（见图 2-20）。此时硬件组态窗口的机架中允许放置该模块的 1 号槽变为绿色，其他插槽为灰色。用鼠标左键按住该模块不放，移动鼠标，将选中的模块“拖”到机架的 1 号槽。

光标没有移动到允许放置该模块的插槽时，其形状为⊘（禁止放置）。拖到 1 号槽时，光标的形状变为，表示允许放置。此时松开鼠标左键，电源模块被放置到 1 号槽。

（2）用双击的方法放置硬件对象

放置模块还有另一个简便的方法，首先用鼠标左键单击机架中需要放置模块的插槽，使它的背景色变为深色。用鼠标左键双击硬件目录中允许放置在该插槽的模块，该模块便出现在选中的插槽，同时自动选中下一个槽。

3．放置信号模块

打开文件夹“\SIMATIC 300\SM-300”，其中的 DI、DO 分别是数字量输入模块和数字量输出模块，AI、AO 分别是模拟量输入模块和模拟量输出模块。

用上述的方法，将 16 点的 DI 模块和 8 点的 DO 模块分别放置在 4 号槽和 5 号槽。

硬件信息显示窗口显示 S7-300 站点中各模块的详细信息，例如模块的订货号、I/O 模块的字节地址和注释等。图 2-20 中 CPU 的固件版本号为 V2.0，MPI 站地址为 2，“DP”行的 2047 是 CPU 集成的 PROFIBUS-DP 接口的诊断地址。

S7-300 的数字量（或称开关量）地址由地址标识符、地址的字节部分和位部分组成，一个字节由 0～7 这 8 位组成。地址标识符 I 表示输入，Q 表示输出，M 表示位存储器。例如 I1.2 是一个数字量输入点的地址，小数点前面的 1 是地址的字节部分，小数点后面的 2 表示这个输入点是 1 号字节中的第 2 位。数字量输入点 I2.0～I2.7 组成了输入字节 IB2。

S7-300 的信号模块的字节地址与模块所在的机架号和插槽号有关。从 0 号字节开始，在组态时 STEP 7 自动地为每个数字量信号模块保留 4B（4 个字节）的地址，相当于 32 个数字量 I/O 点。分配给 4 号槽的 DI 模块的地址为 IB0 和 IB1（见图 2-20），分配给 5 号槽的 DO 模块的地址为 QB4。

模拟量模块以通道为单位，一个通道占用一个字或两个字节的地址。S7-300 的模拟量模块的地址范围为 IB256～767。一个模拟量模块最多 8 个通道，S7-300 给每个模拟量模块自动分配 16B（8 个字）的地址。如果 4 号槽放置的是模拟量模块，其地址从 256 号字节开始。图 2-20 中 6 号槽的 AI 模块的起始地址为 288（256＋2×16），7 号槽的 AO 模块的起始地址为 304（256＋3×16）。

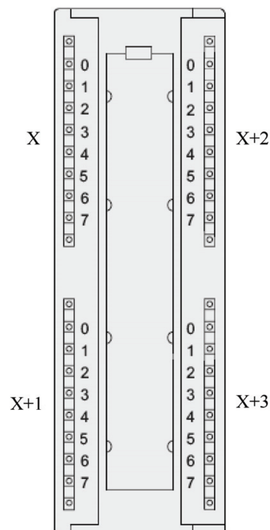


图 2-21　信号模块的地址

模块内各 I/O 点的位地址与信号线接在模块上的哪一个端子有关。图 2-21 是一块 32 点数字量 I/O 模块，其起始字节地址为 X，

每个字节由 8 个 I/O 点组成。图中标出了各 I/O 点字节的位置和字节内各点的端子位置。信号模块内部的地址分配可查阅有关的手册。

4．设置 DI 模块的参数

双击 4 号槽的 16 点 DI 模块，打开它的属性对话框（见图 2-22），“常规”选项卡是模块的基本信息。单击“地址”选项卡中的“系统默认”复选框，其中的“ √ ”消失，“开始”文本框的背景由灰色变为白色，可以用它修改模块的起始地址。建议采用 STEP 7 自动分配的模块地址，不要修改它们，但是在编程时必须使用组态时分配的地址。

在“输入”选项卡（见图 2-22 的右图），用鼠标单击复选框，可以设置是否启用诊断中断和硬件中断。复选框内出现“√”表示允许产生中断。低档的 DI 模块的属性对话框没有“输入”选项卡。

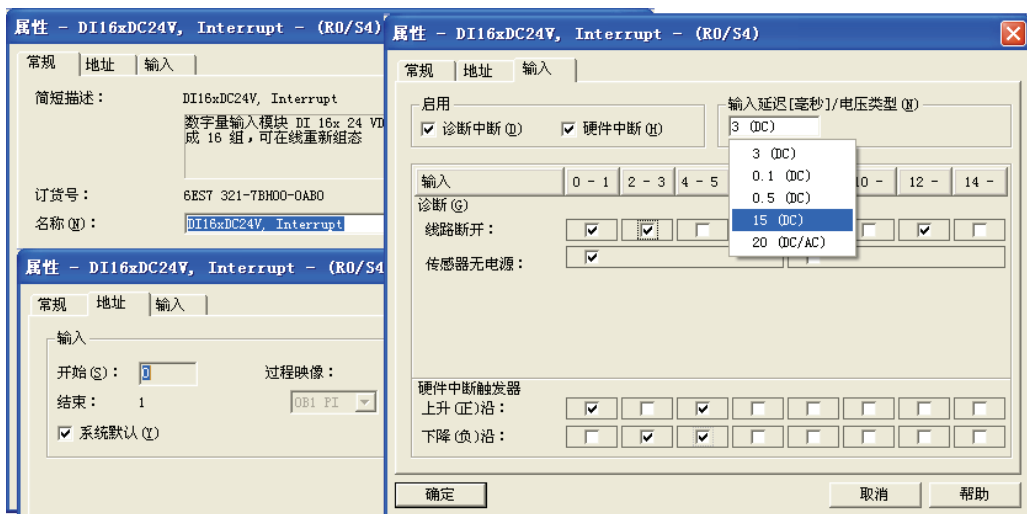


图 2-22　数字量输入模块的参数设置

选中了“硬件中断”复选框后，可以用“硬件中断触发器”区的复选框，设置上升沿中断、下降沿中断，或上升沿和下降沿均产生中断。出现硬件中断时，CPU 的操作系统将调用硬件中断组织块 OB40。

在“诊断”区，可以分组设置是否有断线诊断功能，以及是否诊断传感器电源丢失。如果激活了诊断中断，上述故障事件发生时，CPU 将调用诊断中断组织块 OB82。

机械触点接通和断开时，由于触点的抖动，实际的波形如图 2-23 所示。这样的波形可能会影响程序的正常执行，例如扳动一次开关，使计数器多次计数。可以用输入延迟来滤除图中的窄脉冲。单击“输入延迟”下拉式列表框，在弹出的菜单中选择以 ms 为单位的整个模块的输入延迟时间。输入延迟越大，滤波的效果越好。



图 2-23　波形图

单击选项卡中的“帮助”按钮或按计算机的<F1>键，可以打开该选项卡的在线帮助文件。

5．设置 DO 模块的参数

双击 5 号槽的 8 点 DO 模块，打开它的属性对话框。选中“输出”选项卡的“诊断中



断”复选框（见图 2-24），可以在“诊断”区逐点设置是否有断线、丢失负载电压 L+、对 M 点短路和对 L+点短路的诊断功能。低档的 DO 模块的属性对话框没有“输出”选项卡。

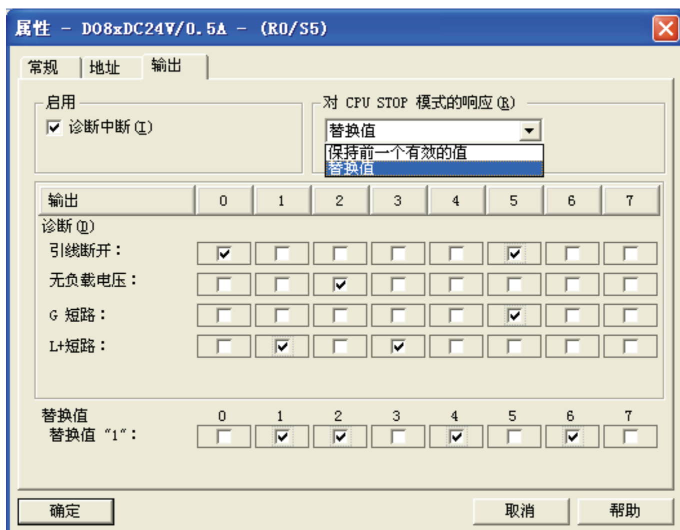


图 2-24　数字量输出模块的参数设置

“对 CPU STOP 模式的响应”选择框用来选择 CPU 进入 STOP 模式时，模块各输出点的处理方式。如果选中“保持前一个有效的值”，进入 STOP 模式后，模块将保持最后的输出值。

如果选中“替换值”，CPU 进入 STOP 模式后，可以使各输出点分别输出“0”或“1”。在对话框下面的“替换值”区的“替换值‘1’”所在的行，为每个输出点设置替换值。多选框内出现“√”表示 CPU 进入 STOP 后该点为 1，反之为 0。

6．设置 AI 模块的参数

双击 HW Config 的机架中订货号为 6ES7-331-7KF02-0AB0 的 8 通道 12 位模拟量输入模块，模块的参数主要在“输入”选项卡（见图 2-25）中设置。



图 2-25　模拟量输入模块的参数设置

（1）测量范围的选择

图 2-25 中每两个通道为一组，可以选择每一通道组的量程。单击“测量型号”输入框，在弹出的菜单中选择测量的种类，测量种类符号的意义见表 2-2。

表 2-2　模拟量输入模块测量种类的符号

符　　号	意　　义	符　　号	意　　义
E	电压	TC-L	热电偶（线性）
4DMU	电流（4 线制变送器）	TC-I	热电偶（内部比较）
2DMU	电流（2 线制变送器）	TC-E	热电偶（外部比较）
R-4L	电阻（4 线连接）	TC-IL	热电偶（线性，内部比较）
R-3L	电阻（3 线连接）	TC-EL	热电偶（线性，外部比较）
RTD-4L	热电阻（线性，4 线连接）	TC-L00C	热电偶（线性，参考温度 0℃）
RTD-3L	热电阻（线性，3 线连接）	TC-L50C	热电偶（线性，参考温度 50℃）

单击“测量范围”输入框，在弹出的菜单中选择量程，图中第一组的测量范围为 4～20mA。测量范围输入框下面的“[C]”表示 0 号和 1 号通道对应的量程卡的位置应设置为“C”（见图 2-25），即量程卡上的“C”旁边的三角形箭头应对准 AI 模块上的箭头。组态好测量范围后，应保证量程卡的位置与组态时要求的位置一致。

如果未使用某一组的通道，应选择测量种类中的“取消激活”，禁止使用该通道组，以减小模块的扫描时间。

（2）模块测量精度与转换时间的设置

SM 331 系列 AI 模块采用积分式 A/D 转换器，积分时间与干扰抑制频率互为倒数。积分时间、干扰抑制频率、转换精度和基本转换时间的关系如表 2-3 所示。积分时间越长，精度越高，快速性越差。积分时间为 20ms 时，对 50Hz 的干扰噪声有很强的抑制作用。为了抑制工频信号对模拟量信号的干扰，一般选择积分时间为 20ms。

单击图 2-25 最右边的“积分时间”所在的方框，用弹出的菜单选择按积分时间或按干扰抑制频率来设置参数。单击各组的积分时间文本框，用弹出的菜单选择需要的参数。

表 2-3　6ES7 331 模拟量输入模块的参数

参　　数	数　　据			
积分时间/ms	2.5	16.6	20	100
干扰抑制频率/Hz	400	60	50	10
转换精度/bit (包括符号位)	9	12	12	14
基本转换时间/ms (包括积分时间)	3	17	22	102

（3）模块诊断与中断功能的设置

可以设置是否允许诊断中断，用多选框设置各组是否有组诊断功能和断线检查功能。模拟量输入模块在出现下列故障时发出诊断消息：外部辅助电源故障、组态/参数设置出错、共模错误、断线、下溢出和上溢出。

如果选择了模拟值超出限制值的硬件中断，“上限”和“下限”输入框的背景变为白色，可以设置产生超限中断的上限值和下限值。

7．设置 AO 模块的参数

如果设置了启用“诊断中断”（见图 2-26），模拟量输出模块无外部负载电压、有组态/



编程错误、对 M 点短路或断线时，将发出诊断消息，触发诊断中断，CPU 将调用 OB 82。

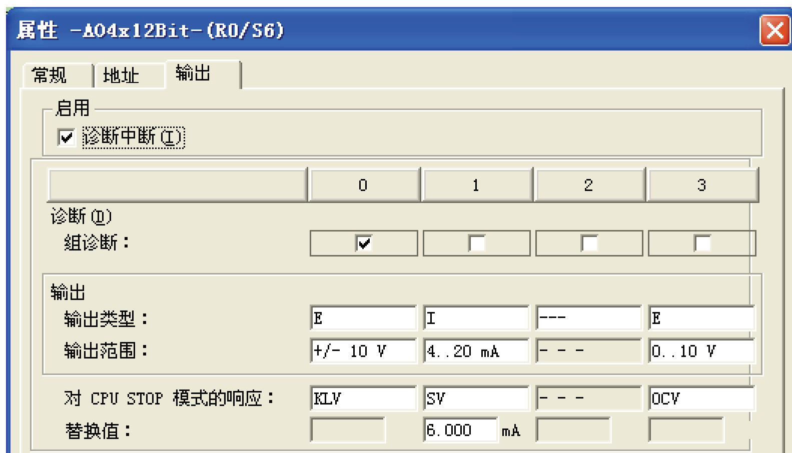


图 2-26　模拟量输出模块的参数设置

输出类型可选取消激活、电压输出或电流输出。选定输出类型后，再选择输出信号的量程（输出范围）。

CPU 进入 STOP 模式时，可以选择不输出电流电压（0CV）、保持最后的输出值（KLV）和采用替换值（SV）。选择 SV 后，应设置替换值的大小。

8．编译和保存组态信息

组态结束后，单击工具栏上的按钮（编译并保存）。编译成功后，选中 SIMATIC 管理器左边窗口最下面的“块”，右边窗口可以看到编译后生成的保存硬件组态信息和网络组态信息的“系统数据”图标。单击 SIMATIC 管理器工具栏的下载按钮，可以将它下载到 CPU，也可以在 HW Config 中将硬件组态信息下载到 CPU。

9．硬件组态练习

用新建项目向导生成一个项目，选用 S7-400 的 CPU。

打开 HW Config，将电源模块和信号模块插入机架。设置 DI 模块、DO 模块、AI 模块和 AO 模块的参数。

2.3　实训三　异步电动机正反转控制

2.3.1　生成用户程序

1．硬件电路

图 2-27 是三相异步电动机正反转控制的主电路和继电器控制电路，KM1 和 KM2 分别是控制正转运行和反转运行的交流接触器。图中的 FR 是用于过载保护的热继电器。

图 2-28 是 PLC 的外部接线图和梯形图，各输入信号均用常开触点提供。输出电路中的硬件互锁电路用于确保 KM1 和 KM2 的线圈不会同时通电，以防止出现交流电源相间短路的故障。

2．生成项目

用“新建项目”向导生成一个名为“电机控制”的项目，CPU 可以选任意的型号。如果

只是用于仿真实验，可以不对 S7-300 的硬件组态，机架中只有 CPU 模块也能仿真。如果使用 S7-400 的 CPU，必须组态电源模块才能进行仿真。

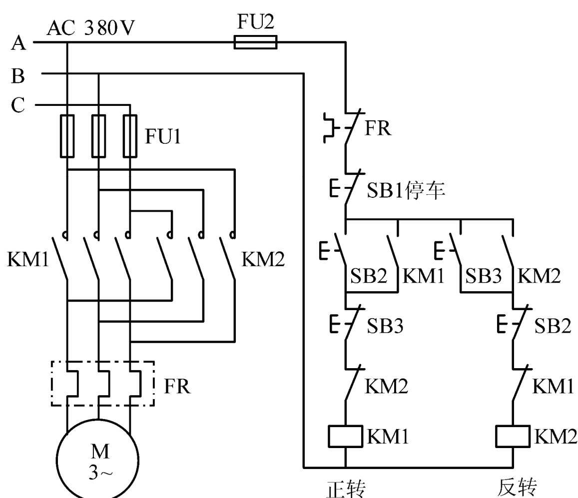


图 2-27　异步电动机正反转控制电路图

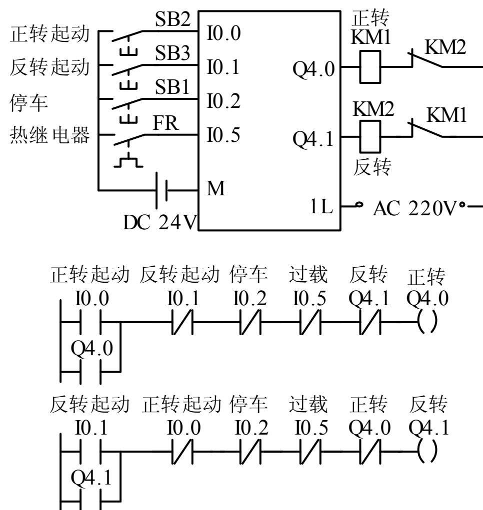


图 2-28　PLC 外部接线图与梯形图

3．定义符号地址

在程序中可以用绝对地址（例如 I0.2）访问变量，但是符号地址（例如“停止按钮”）使程序更容易阅读和理解。用符号表定义的符号可供所有的逻辑块使用。

选中 SIMATIC 管理器左边窗口的“S7 程序”，双击右边窗口出现的“符号”，打开符号编辑器（见图 2-29），OB1 的符号是自动生成。在下面的空白行输入符号“正转按钮”和地址 I0.0，其数据类型 BOOL(二进制的位)是自动添加的。可以为符号输入注释。

单击某一列的表头，可以改变排序的方法。例如单击“地址”所在的单元，该单元出现向上的三角形，表中的各行按地址升序排列（按地址的第 1 个字母从 A 到 Z 的顺序排列）。再单击一次“地址”所在的单元，该单元出现向下的三角形，表中的各行按地址降序排列。

符号编辑器 - [S7 程序(1) (符号) -- S7_Pro1\SI...

符号表(S)　编辑(E)　插入(I)　视图(V)　选项(O)　窗口(W)　帮助(H)

全部符号

	状态	符号	地址		数据类型		注释
1		正转按钮	I	0.0	BOOL		常开触点
2		反转按钮	I	0.1	BOOL		常开触点
3		停车按钮	I	0.2	BOOL		常开触点
4		过载	I	0.5	BOOL		常开触点
5		Cycle Execution	OB	1	OB	1	
6		电机正转	Q	4.0	BOOL		
7		电机反转	Q	4.1	BOOL		

图 2-29　符号表

4．生成梯形图程序

选中 SIMATIC 管理器左边窗口中的“块”（见图 2-18），双击右边窗口中的 OB1，打开程序编辑器（见图 2-30）。

第一次打开程序编辑器时，程序块和每个程序段均有灰色背景的注释区。注释区比较占



地方，可以执行菜单命令“视图”→“显示方法”→“注释”，关闭所有的注释区。下一次打开该程序块后，需要做同样的操作来关闭注释。

执行下面的操作，可以在打开程序块时不显示注释区：在程序编辑器中执行菜单命令“选项”→“自定义”，在打开的“自定义”对话框的“视图”选项卡中（见图 2-30 中间的图），取消“块打开后的视图”区中对“块/程序段注释”的激活，即用鼠标单击它左边的复选框，使其中的“√”消失。

在“自定义”对话框的“LAD/FBD”选项卡（见图 2-30 右边的图），可以设置“地址域宽度”，即梯形图中触点和线圈的宽度（以字符个数为单位）。

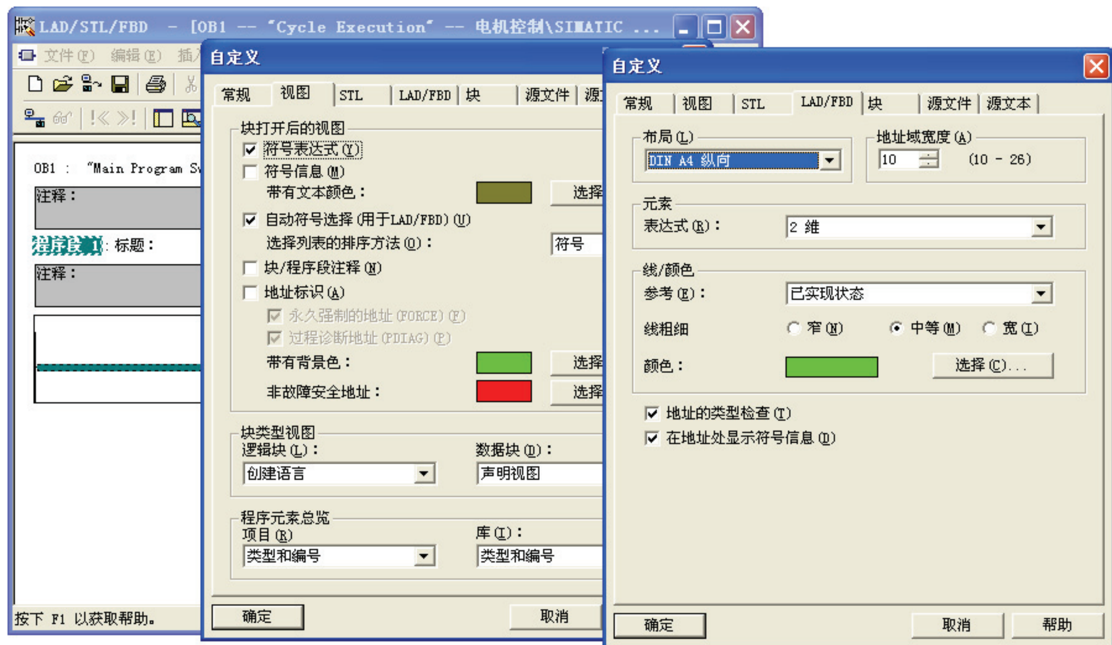


图 2-30　自定义程序编辑器的属性

关闭程序段的注释后，可以将程序段的简要注释放在程序段的“标题”行。

如果在新建项目时，图 2-17 的左图选中的是默认的“STL”(语句表)，打开程序编辑器后，只能输入语句表。此时需要执行菜单命令“视图”→“LAD”，将编程语言切换为梯形图。

单击程序段 1 梯形图的水平线，它变为深色的加粗线（见图 2-30 的左图）。

单击一次工具栏上的常开触点按钮，单击 4 次常闭触点按钮，单击一次线圈按钮，生成的触点和线圈见图 2-31a。

为了生成并联的触点，首先单击最左边的垂直短线来选中它，然后单击工具栏上的按钮，生成一个常开触点（见图 2-31b）。单击工具栏上的按钮，该触点被并联到上面一行的第一个触点上（见图 2-31c）。

用鼠标右键单击触点上的“??.?”，执行弹出的快捷菜单中的“插入符号”命令（见图 2-31d），打开下拉式符号表（见图 2-31e），双击其中的变量“电机正转”，该符号地址出现在触点上。用同样的方法输入其他符号地址。

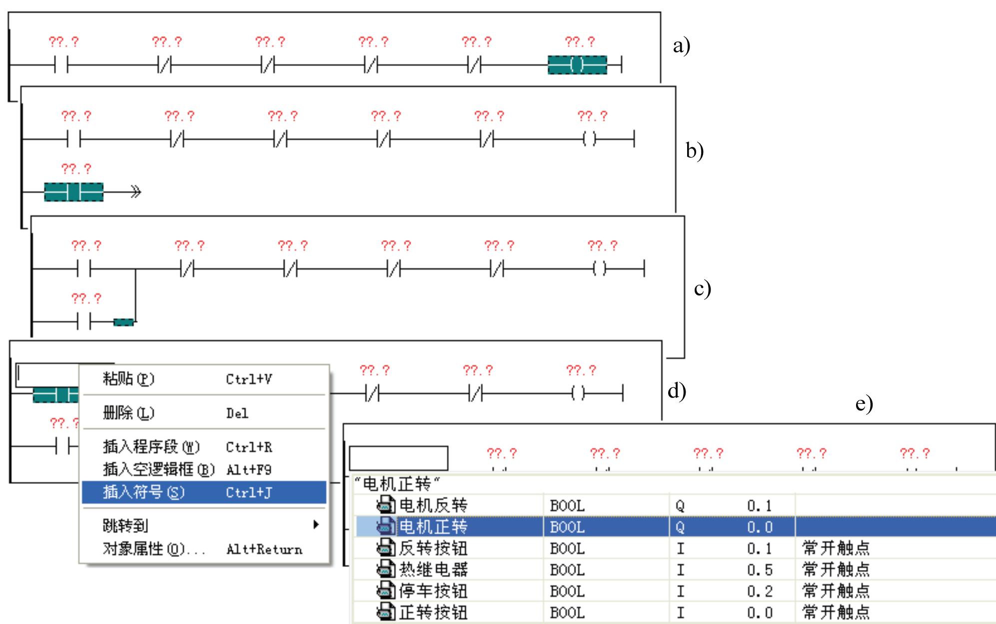


图 2-31　生成用户程序

图 2-32 是输入结束后的梯形图，STEP 7 自动地为程序中的全局符号加双引号。

STEP 7 的鼠标右键功能是很强的，用右键单击窗口中的某一对象，在弹出的快捷菜单中将会出现与该对象有关的最常用的命令。单击某一菜单项，可以执行相应的操作。建议在使用软件的过程中逐渐熟悉和使用右键功能。

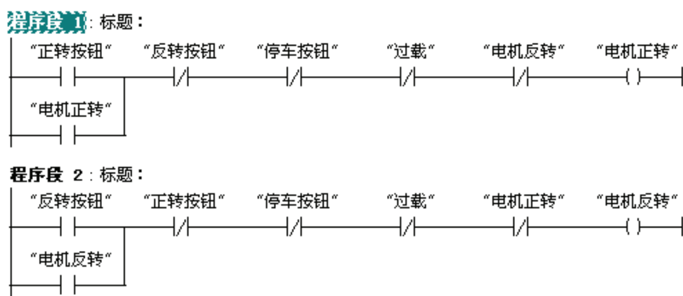


图 2-32　梯形图程序

执行菜单命令“视图”→“显示方式”→“符号表达式”，菜单中该命令左边的符号“√”消失，梯形图中的符号地址变为绝对地址。再次执行该命令，该命令左边出现“√”，又显示符号地址。

执行菜单命令“视图”→“显示方式”→“符号信息”，在符号地址的上面出现绝对地址和符号表中的注释（见图 2-33），菜单中该命令的左边出现符号“√”。再次执行该命令，该命令左边的“√”消失，只显示符号地址。

用鼠标左键选中双箭头表示的触点的端点，按住左键不放，将自动出现的与端点连接的线拖到希望并允许放置的位置，随光标一起移动的⊘（禁止放置）符号变为┓┏（允许放置）时（见图 2-34），放开左键，该触点便被连接到指定的位置。



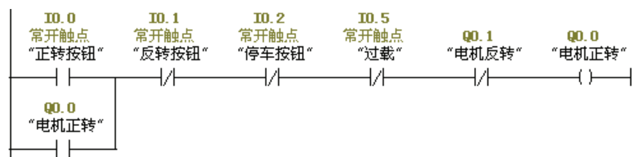


图 2-33　显示符号信息的梯形图程序

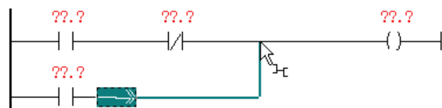


图 2-34　梯形图中触点的并联

2.3.2　用 PLCSIM 调试程序

1．打开仿真软件 PLCSIM

S7-PLCSIM 是 S7-300/400 功能强大、使用方便的仿真软件。可以用它代替 PLC 硬件来调试用户程序。

安装 PLCSIM 后，SIMATIC 管理器工具栏上的按钮由灰色变为深色。单击该按钮，第一次打开 PLCSIM 时，出现图 2-35 所示的对话框，选中文本框中的“SIMATIC S7 PLCSIM”，“Activate”（激活）按钮上的字符颜色变为黑色，单击它将激活 14 天的试用许可证密钥。

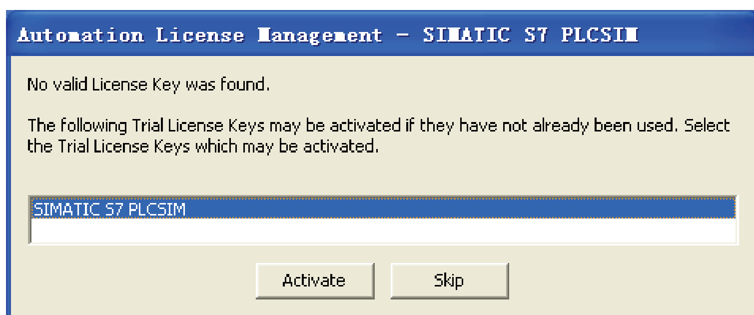


图 2-35　激活试用许可证密钥

打开 S7-PLCSIM 后，自动建立了 STEP 7 与仿真 CPU 的 MPI 连接。刚打开 PLCSIM 时，只有图 2-36 最左边被称为 CPU 视图对象的小方框。单击它上面的“STOP”、“RUN”或“RUN-P”小方框，可以令仿真 PLC 处于相应的运行模式。单击“MRES”按钮，可以清除仿真 PLC 中已下载的程序。

可以用鼠标调节 S7-PLCSIM 窗口的位置和大小。还可以执行菜单命令“View”→“Status Bar”，关闭或打开下面的状态条。

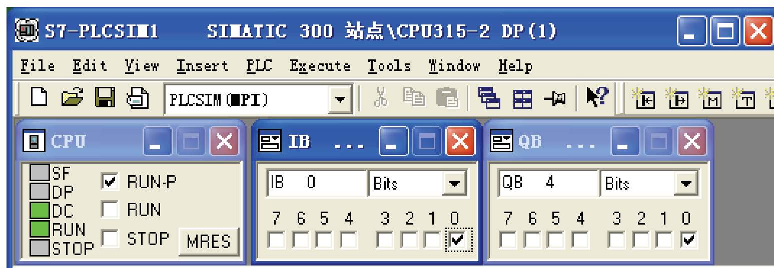


图 2-36　PLCSIM

单击工具栏上的按钮后，如果不能打开 PLCSIM，根据作者的经验，与 Windows 操作系统有关，应更换操作系统。

2．下载用户程序和组态信息

单击 S7-PLCSIM 工具栏上的和按钮，生成 IB0 和 QB0 视图对象。将视图对象中的 QB0 改为 QB4（见图 2-36），按计算机的〈Enter〉键后更改才生效。

下载之前，应打开 PLCSIM。选中 SIMATIC 管理器左边窗口中的“块”对象，单击工具栏的下载按钮，将 OB1 和系统数据下载到仿真 PLC。下载系统数据时出现“是否要装载系统数据？”对话框时，单击“是”按钮确认。

不能在 RUN 模式时下载，但是可以在 RUN-P 模式下载。在 RUN-P 模式下载系统数据时，将会出现“模块将被设为 STOP 模式？”的对话框。下载结束后，出现“是否现在就要启动该模块？”的对话框。单击“是”按钮确认。

3．用 PLCSIM 的视图对象调试程序

单击 CPU 视图对象中的小方框，将 CPU 切换到 RUN 或 RUN-P 模式。这两种模式都要执行用户程序，但是在 RUN-P 模式可以下载修改后的程序块和系统数据。

根据梯形图电路，按下面的步骤调试用户程序：

1）单击视图对象 IB0 最右边的小方框，方框中出现“√”，I0.0 变为 1 状态，模拟按下正转按钮。梯形图中 I0.0 的常开触点闭合、常闭触点断开。由于 OB1 中程序的作用，Q4.0（电机正转）变为 1 状态，梯形图中其线圈通电，视图对象 QB4 最右边的小方框中出现“√”（见图 2-36）。

再次单击 I0.0 对应的小方框，方框中的“√”消失，I0.0 变为 0 状态，模拟放开起动按钮。梯形图中 I0.0 的常开触点断开、常闭触点闭合。将按钮对应的位（例如 I0.0）设置为 1 之后，注意一定要马上将它设置为 0，否则后续的操作可能会出现异常情况。

2）单击两次 I0.1 对应的小方框，模拟按下和放开反转起动按钮的操作。由于用户程序的作用，Q4.0 变为 0 状态，Q4.1 变为 1 状态，电动机由正转变为反转。

3）在电动机运行时用鼠标模拟按下和放开停止按钮 I0.2，或模拟过载信号 I0.5 出现和消失，观察当时处于 1 状态的 Q4.0 或 Q4.1 是否变为 0 状态。

4．下载部分程序块

程序块较多时，可以只下载部分程序块。打开随书光盘中的项目“S7_DP”，选中左边窗口的“块”文件夹，单击右边窗口的某个块或系统数据，被选中的块的背景色变为深蓝色。打开 PLCSIM，单击工具栏的下载按钮，只下载选中的对象。图 2-37 中的“VAT_1”是用于监控程序执行情况的变量表，即使选中它也不会下载它。

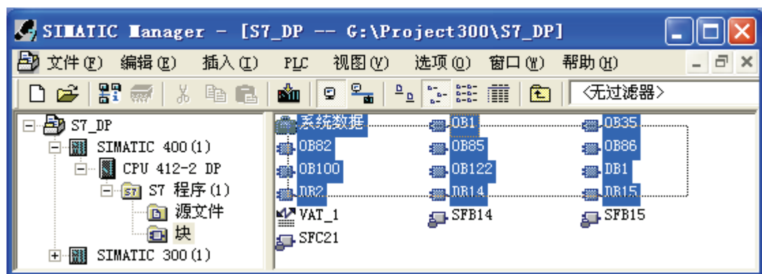


图 2-37　选择需要下载的块



用鼠标左键单击图 2-37 中块工作区的某一点，按住左键不放，移动鼠标，画出一个虚线方框，方框内的块被选中。单击工具栏的下载按钮，只下载选中的对象。

按住计算机的〈Ctrl〉键，单击需要下载的块，可以选中多个任意位置的块。单击工具栏的下载按钮，只下载选中的对象。

修改程序后，也可以在程序编辑器中下载打开的程序块。

5. 下载整个站点

选中项目中的某个 PLC 站点，单击工具栏的下载按钮，可以把整个站点的信息（包括程序块、系统数据中的硬件组态和网络组态信息）下载到 CPU 中。

6. 在线窗口与离线窗口

单击工具栏上的在线按钮，将打开在线窗口（见图 2-38）。该窗口最上面的标题栏出现浅蓝色背景的长条，表示在线。如果选中管理器左边窗口中的“块”，右边的窗口将会列出 CPU 集成的大量的系统功能块 SFB、系统功能 SFC，和已经下载到 CPU 的系统数据和用户编写的块。SFB 和 SFC 在 CPU 的操作系统中，无需下载，也不能用编程软件删除。在线窗口显示的是 PLC 中的内容，而离线窗口显示的是计算机中的内容。

打开在线窗口后，可以用 SIMATIC 管理器工具栏上的按钮和按钮，或者用管理器的“窗口”菜单来切换在线窗口和离线窗口。单击右上角的按钮，关闭在线窗口后，离线窗口仍然存在。



图 2-38　在线窗口

打开在线窗口后，执行菜单命令“窗口”→“排列”→“水平”，将会同时显示在线窗口和离线窗口。可以用拖放的方法，将离线窗口中的块拖到在线窗口的块工作区（下载块）。也可以将在线窗口中的块拖到离线窗口的块工作区（上载块）。

7. 用程序状态功能调试程序

仿真 CPU 在 RUN 或 RUN-P 模式时，打开 OB1，单击工具栏上的“监视”按钮，启动程序状态监控功能。

STEP 7 和 PLC 中的 OB1 程序不一致时（例如下载后改动了程序），工具栏的按钮上的符号为灰色。此时需要单击工具栏的下载按钮，重新下载 OB1。STEP 7 和 PLC 中 OB1 的程序一致后，按钮上的符号变为黑色，才能启动程序状态功能。

从梯形图左侧垂直的“电源”线开始的水平线均为绿色（见图 2-39），表示有能流从

"电源"线流出。有能流流过的方框指令、线圈和"导线"，以及处于闭合状态的触点，均用绿色表示。用蓝色虚线表示没有能流流过和触点、线圈断开。

如果选中程序段 2，只能监控程序段 2 和它之后的程序段，不能监控程序段 1。

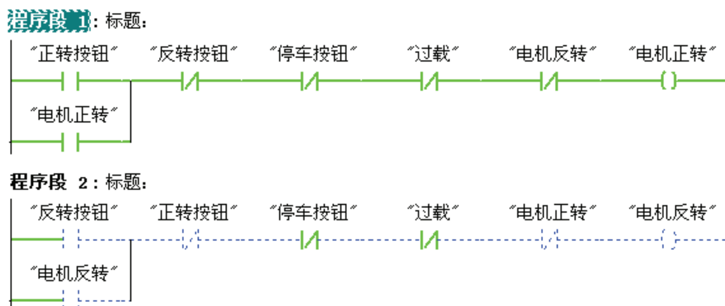


图 2-39　程序状态监控

8．在 PLCSIM 中使用符号地址

执行菜单命令"Tools"→"Options"→"Attach Symbols"（连接符号），单击打开的对话框中的"浏览"按钮（见图 2-40），选中要仿真的项目"电机控制"。打开项目中的 300 站点，选中"S7 程序"，单击右边窗口的"符号"，在"对象名称"文本框中出现"符号"。单击"确定"按钮退出对话框。

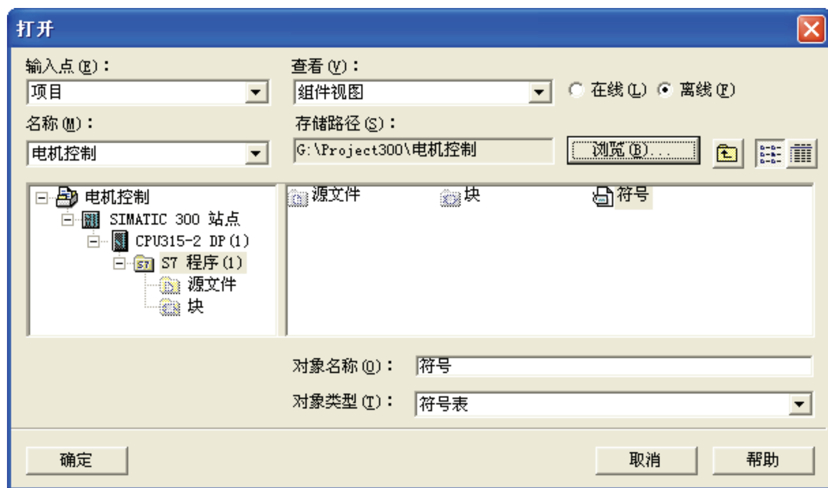


图 2-40　连接符号表

执行菜单命令"Tools"→"Options"→"Show Symbols"（显示符号），使该指令项的左边出现"√"（被选中）。单击工具栏上的按钮，打开垂直位列表（Vertical Bits）视图对象。设置它的地址为 IB0，视图对象的下面显示 IB0 中已定义的符号地址（见图 2-41）。

单击工具栏上的按钮，打开堆栈（Stack）视图对象（见图 2-41），里面有 Nesting Stack（嵌套堆栈）和 MCR（主控继电器）堆栈。



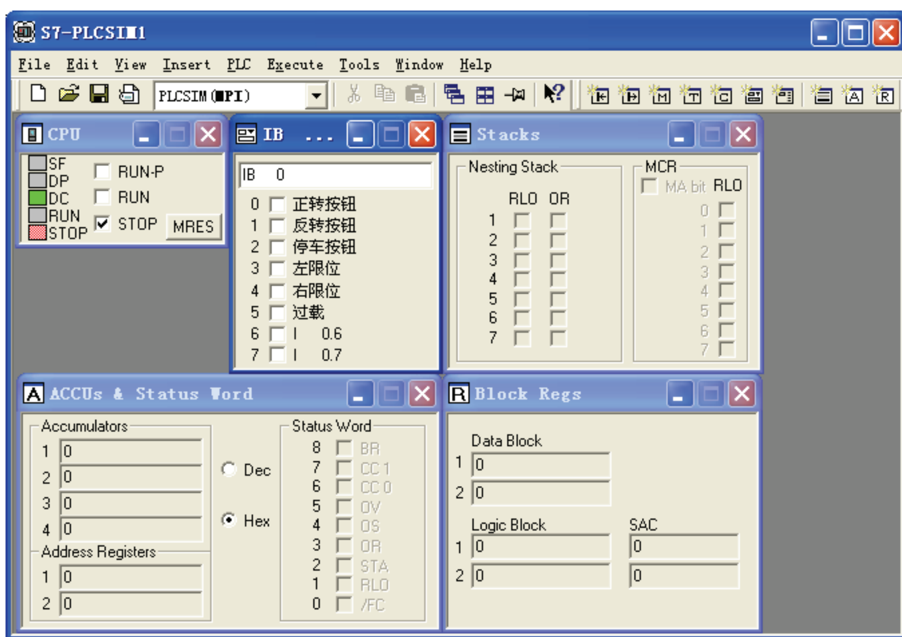


图 2-41 PLCSIM 的视图对象

单击工具栏上的按钮，打开累加器与状态字（ACCUS & Status Word）视图对象。可以监控累加器（Accumulators）、地址寄存器（Address Registers）和状态字（Status Word）。

单击工具栏上的按钮，打开块寄存器（Block Regs）视图对象，可以监控数据块地址寄存器（Data Block address registers）、逻辑块（Logic Block）的编号和步地址计数器 SAC（Step address counter）。实际上很少使用堆栈视图对象和块寄存器视图对象。

9. 仿真练习

用新建项目向导生成一个项目，用起动按钮和停止按钮控制一台单向运行的电动机，电机过载时自动停机。画出 PLC 的外部接线图。

用符号表定义输入、输出变量的符号，生成梯形图程序。用 PLCSIM 和程序状态监控功能调试程序，在 PLCSIM 中使用符号地址。

2.4 实训四 小车控制系统

2.4.1 PLC 的循环处理过程

1. PLC 的循环处理过程

CPU 的程序分为操作系统和用户程序。操作系统用来处理 PLC 的启动、刷新过程映像输入/输出区、调用用户程序、处理中断和错误、管理存储区和通信等任务。

用户程序由用户生成，用来实现用户要求的自动化任务。

PLC 得电或从 STOP 模式切换到 RUN 模式时，CPU 执行启动操作，清除没有保持功能

的位存储器、定时器和计数器，清除中断堆栈和块堆栈的内容，复位保存的硬件中断等。此外还要执行一次用户生成的“系统启动”组织块OB100，完成用户指定的初始化操作。以后 PLC 采用循环执行用户程序的方式，这种运行方式也称为扫描工作方式。

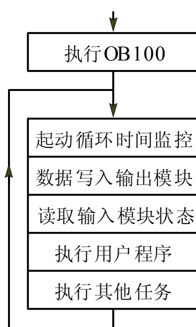


图 2-42　扫描过程

在 PLC 的存储器中，设置了一片区域用来存放输入信号和输出信号的状态，它们分别称为过程映像输入区和过程映像输出区。PLC 梯形图中的其他编程元件也有对应的存储区。

下面是循环处理各个阶段的任务（见图 2-42）：

1）操作系统启动循环时间监控。

2）CPU 将过程映像输出区（Q 区）的数据写到输出模块。

3）CPU 读取输入模块的输入状态，并存入过程映像输入区（I 区）。

4）CPU 处理用户程序，执行用户程序中的指令。

5）在循环结束时，操作系统执行所有挂起的任务，例如下载和删除块，接收和发送全局数据等。

6）CPU 返回第一阶段，重新启动循环时间监控。

在启动完成后，不断地循环调用组织块 OB1，OB1 是用户程序中的主程序，它可以调用别的程序块。

PLC 的用户程序由若干条指令组成，指令在存储器中顺序排列。在没有跳转指令和块调用指令时，CPU 从第一条指令开始，逐条顺序地执行用户程序，直到用户程序结束之处。在执行指令时，从过程映像输入区或别的存储区中将有关编程元件的 0、1 状态读出来，并根据指令的要求执行相应的逻辑运算，运算的结果写入到对应的存储单元，因此，各编程元件的存储区 (过程映像输入区除外)的内容随着程序的执行而变化。

循环程序处理过程可以被某些事件中断。如果有中断事件出现，当前正在执行的块被暂停执行，并自动调用分配给该事件的组织块。该组织块被执行完后，被暂停执行的块将从被中断的地方开始继续执行。

2．过程映像输入/输出区

在循环程序处理过程中，CPU 并不直接访问 I/O 模块中的输入地址区和输出地址区，而是访问 CPU 内部的过程映像区（I/Q 区）。

在写输出模块阶段，CPU 将过程映像输出区的状态传送到输出模块。梯形图中某一数字量输出位（例如 Q4.0）的线圈“通电”时，对应的过程映像输出位为 1 状态。信号经输出模块隔离和功率放大后，继电器型输出模块对应的硬件继电器的线圈通电，其常开触点闭合，使外部负载通电工作。

若梯形图中输出位的线圈“断电”，对应的过程映像输出位为 0 状态，在写输出模块阶段之后，继电器型输出模块对应的硬件继电器的线圈断电，其常开触点断开，外部负载断电，停止工作。

在读输入模块阶段，PLC 把所有外部输入电路的接通/断开状态读入过程映像输入区。

外部输入电路接通时，对应的过程映像输入位(例如 I0.0)为 1 状态，梯形图中该输入位的常开触点接通，常闭触点断开。外部输入电路断开时，对应的过程映像输入位为 0 状态，梯形图中该输入位的常开触点断开，常闭触点接通。



某一编程元件对应的位为 1 状态时，称该编程元件为 ON，该位为 0 状态时，称该编程元件为 OFF。

在程序执行阶段，即使外部输入电路的状态发生了变化，过程映像输入位的状态也不会随之而变，输入信号变化了的状态只能在下一个扫描循环周期的读取输入模块阶段被读入。

扫描周期（Scan Cycle）是指操作系统执行一次如图 2-42 所示的循环操作所需的时间，扫描周期又称为扫描循环时间。

3．外设输入/外设输出区

S7-300/400 的外设输入/外设输出区（PI/PQ 区）用于直接读写 I/O 模块。

PI/PQ 区与 I/Q 区的关系如下：

1）访问 PI/PQ 区时，直接读写输入/输出模块，而 I/Q 区是输入/输出信号在 CPU 的存储区中的“映像”。

2）I/Q 区可以按位、字节、字和双字访问，PI/PQ 区不能按位访问。

3）I/Q 区的地址范围比 PI/PQ 区的小（前者与 CPU 的型号有关）。I/Q 区的地址也可以用 PI/PQ 区访问。如果地址超出了 I/Q 区允许的范围，必须使用 PI/PQ 区来访问。

2.4.2　小车控制系统

1．控制系统简介

图 2-43 是小车控制系统的示意图与外部接线图。按下右行起动按钮 SB2 或左行起动按钮 SB3 后，要求小车在左限位开关 SQ1 和右限位开关 SQ2 之间不停地循环往返，直到按下停止按钮 SB1。

2．仿真实验要求

用“新建项目”向导生成一个名为“小车控制 1”的项目（见随书光盘中的同名例程），CPU 可以选用任意的型号。

打开 OB1，输入图 2-44 所示的梯形图程序。打开 PLCSIM，将用户程序和系统数据下载到仿真 PLC。将仿真 PLC 切换到 RUN 或 RUN-P 模式。生成视图对象 IB0 和 QB4（见图 2-36），用 PLCSIM 调试程序。单击 I0.0～I0.5 对应的小方框，生成各种输入信号，通过观察 Q4.0 和 Q4.1 对应的小方框，检查程序运行的情况。

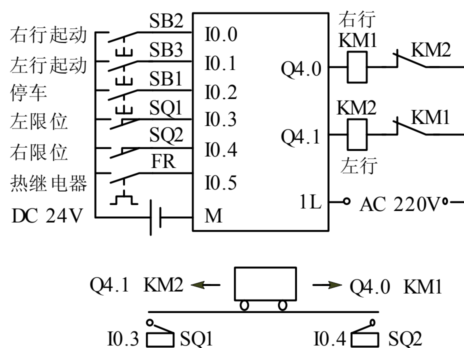


图 2-43　PLC 的外部接线图

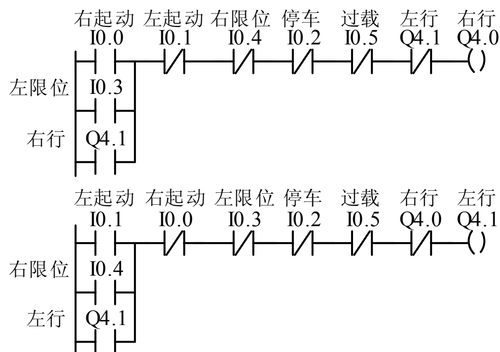


图 2-44　梯形图

打开 OB1，单击程序编辑器工具栏上的“监视”按钮，用程序状态监控功能调试程序。按以下步骤检查程序：

1）开始时所有的输入信号均为 0 状态。单击两次 I0.0 对应的小方框，模拟按下和放开右行起动按钮，观察 Q4.0 是否变为 1 状态。

2）连续单击两次 I0.4 对应的小方框，模拟右限位开关信号，观察 Q4.0 是否变为 0 状态，Q4.1 是否变为 1 状态。

3）连续单击两次 I0.3 对应的小方框，模拟左限位开关信号，观察 Q4.1 是否变为 0 状态，Q4.0 是否变为 1 状态。

4）重复第 2 步和第 3 步。

5）连续单击两次 I0.2 对应的小方框，模拟停车按钮信号，或者连续单击两次 I0.5 对应的小方框，模拟过载信号，观察它们是否能使 Q4.0 或 Q4.1 变为 0 状态。

6）小车在停车状态时，单击 I0.1 对应的小方框，模拟左行起动按钮信号，起动后用上述方法控制小车自动反向运行和停车。

若发现 PLC 的输入/输出关系不符合要求，应检查程序，改正错误。

2.4.3 STEP 7 与 PLC 通信连接的组态

1．USB/MPI 适配器

现在的笔记本电脑和部分台式计算机已经没有 RS-232C 接口，只能使用计算机的 USB 接口和 PLC 通信。USB/MPI 适配器用于连接安装了 STEP 7 的计算机的 USB 接口和 PLC 的 MPI 或 PROFIBUS-DP 接口，特别适合于笔记本电脑使用（见图 2-45）。



图 2-45　PC Adapter USB

2．STEP 7 与 PLC 通信的组态

需要安装随书光盘的文件夹“PC_Adapter_USB”中的驱动程序后，才能使用 USB/MPI 适配器。安装好驱动程序后，在 SIMATIC 管理器中执行菜单命令“选项”→“设置 PG/PC 接口”，打开“设置 PG/PC 接口”对话框（见图 2-46）。在中间的通信接口列表中，选中“PC Adapter（MPI）”，此时适配器应连接到 CPU 的 MPI 接口。计算机一侧的通信速率为 19.2 kbit/s 或 38.4 kbit/s，PLC 一侧的通信速率为 19.2 kbit/s～1.5 Mbit/s。

单击“属性”按钮，打开“属性 – PC Adapter（MPI）”对话框。在“本地连接”选项卡（见图 2-46），用“连接到”选择框设置计算机与 PLC 通信使用的接口为 USB。还需要设置计算机一侧的传输速率。

可以使用“MPI”选项卡中默认的参数（见图 2-46），运行 STEP 7 的计算机在 MPI 网络中默认的站地址为 0。“超时”选择框用来设置与 PLC 建立连接的最长时间。



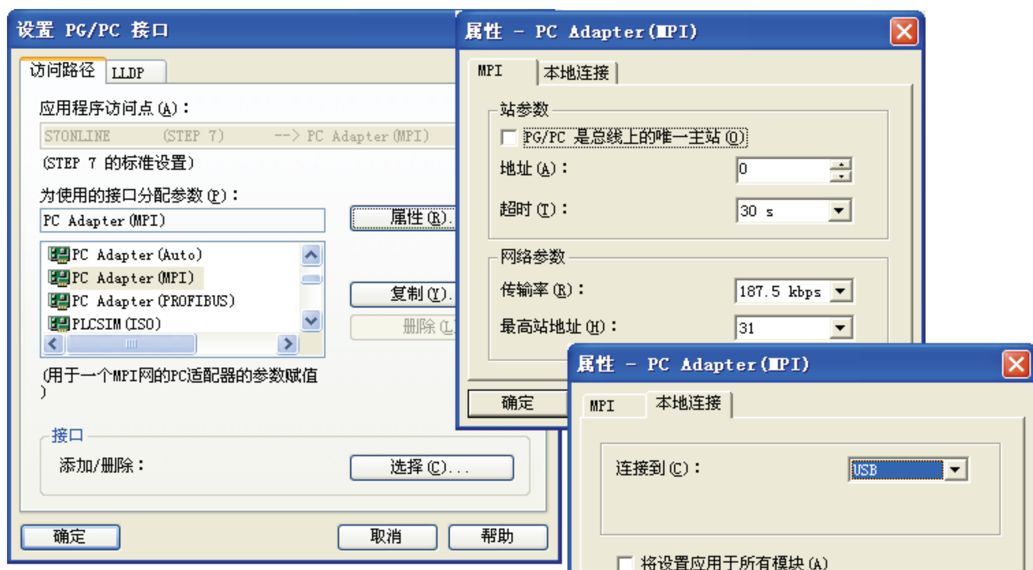


图 2-46　设置 PG/PC 接口对话框

如果选中“设置 PG/PC 接口”对话框中间的通信接口列表的“PC Adapter（PROFIBUS）”，适配器应连接到 CPU 的 DP 接口。更改后单击“确定”按钮，退出对话框后新的设置生效。

3．PC/MPI 适配器

PC/MPI 适配器用于连接计算机的 RS-232C 接口和 PLC 的 MPI 接口或 PROFIBUS-DP 接口。除了 PC 适配器，还需要一根 RS-232C 通信电缆。

在 SIMATIC 管理器中执行菜单命令“选项”→“设置 PG/PC 接口”，打开“设置 PG/PC 接口”对话框（见图 2-46）。单击“选择”按钮，打开“安装/删除接口”对话框（见图 2-47）。



图 2-47　安装/删除接口对话框

选中左边的“选择”列表框中的“PC Adapter”，单击中间的“安装-->”按钮，将安装它的驱动程序，安装好后，“PC Adapter”出现在右边的“已安装”列表框中。

如果要卸载“已安装”列表框中某个已安装的通信硬件的驱动程序，首先选中它，然后单击中间的“<-- 卸载”按钮，该通信硬件在“已安装”列表框中消失，其驱动程序被卸载。单击“关闭”按钮，返回 PG/PC 接口设置对话框。

有的 PC/MPI 适配器上有一个选择传输速率的开关，可选 19.2 kbit/s 或 38.4 kbit/s。组态时设置的传输速率应与适配器硬件设置的传输速率相同。

如果计算机没有 RS-232C 接口，PC/MPI 适配器可以串接一个 RS232/USB 转接器。安装转接器的驱动程序后，在图 2-46 的“属性 - PC Adapter（MPI）”对话框的“本地连接”选项卡（见图 2-46），“连接到”选择框中对应于 RS232/USB 转接器的是一个增加的 COM 口，而不是 USB 接口。

4．安装在计算机内的通信卡

CP 5611、CP 5613、CP 5614 和 CP 5621 是用于台式机的 PCI 卡，CP 5512 是用于笔记本电脑的 PCMCIA 卡。可以用它们来将计算机连接到 MPI 或 PROFIBUS 网络，通过网络实现计算机与 PLC 的通信。

有以太网接口的 CPU 也可以使用计算机的工业以太网通信卡 CP 1512（PCMCIA 卡）或 CP 1613、CP 1616 和 CP 1623（PCI 卡），通过工业以太网实现计算机与 PLC 的通信。

STEP 7 自带 CP 5611、CP 5512 的驱动程序（见图 2-47），需要单独安装其他通信卡的驱动程序，或安装西门子的通信软件 SIMATIC NET。

5．硬件 PLC 的在线操作

完成上述某一种通信硬件的驱动程序的安装和设置，用电缆连接好计算机和 PLC 后，接通它们的电源，就可以进行下载、上传和监控等在线操作了。具体的操作方法和观察到的现象与用 PLCSIM 做仿真实验时的基本上相同。

2.5 练习题

1．填空

1）数字量输入模块某一外部输入电路接通时，对应的过程映像输入位为_____状态，梯形图中对应的常开触点______，常闭触点______。

2）若梯形图中某一过程映像输出位 Q 的线圈“断电”，对应的过程映像输出位为___状态，在写入输出模块阶段之后，继电器型输出模块对应的硬件继电器的线圈______，其常开触点_____，外部负载_____。

3）S7-300 的电源模块在中央机架最___边的 1 号槽，CPU 模块在___号槽，接口模块在___号槽。每个机架最多可安装___个信号模块、功能模块或通信处理器模块。

4）S7-300 中央机架的 4 号槽的 16 点数字量输出模块默认的字节地址为______和______。5 号槽的 32 点数字量输入模块的默认字节地址为______至______。6 号槽的 4AI/2AO 模块的模拟量输入字默认的地址为______至______，模拟量输出字地址为______和______。



5）S7-400 的电源模块必须安装在________号槽。

2．简述 PLC 的循环处理过程。

3．什么是扫描周期？

4．硬件组态的任务是什么？

5．信号模块是哪些模块的总称？

6．怎样在 PLCSIM 中模拟按钮信号的操作？

7．PI/PQ 与 I/Q 有什么区别？位逻辑指令可以使用 PI/PQ 存储区的地址吗？

第 3 章

S7-300/400 的指令应用

3.1 位逻辑指令应用

3.1.1 实训五 位逻辑指令的仿真实验

1. 二进制数

某些物理量只有两种相反的状态，例如电平的高、低，接触器线圈的通电和断电等。它们被称为开关量或数字量。二进制数的 1 位（bit）只能取 0 和 1 这两个不同的值，可以用它们来表示数字量的两种状态。梯形图中的位编程元件（例如存储器位 M 和过程映像输出位 Q）的线圈“通电”时，其常开触点接通，常闭触点断开，以后称该编程元件为 1 状态，或称该编程元件 ON。位编程元件的线圈和触点的状态与上述的相反时，称该编程元件为 0 状态，或称该编程元件 OFF。

位数据的数据类型为 BOOL（布尔）型，在编程软件中，位编程元件的 1 状态和 0 状态分别用 TRUE（真）和 FALSE（假）来表示。

位存储单元的地址由字节地址和位地址组成，例如 I3.2 中的区域标示符“I”表示输入，字节地址为 3，位地址为 2。输入字节 IB3 由 I3.0～I3.7 这 8 位组成。

梯形图中触点的串连、并联可以实现“与”运算与“或”运算，用常闭触点控制线圈可以实现“非”运算（见图 3-1）。用多个触点的串、并联电路可以实现复杂的逻辑运算。“与”、“或”、“非”逻辑运算的输入/输出关系如表 3-1 所示，逻辑运算表达式中的“*”和加号分别表示“与”运算和“或”运算。$\overline{\text{I0.4}}$ 表示对 I0.4 作“非”运算。

表 3-1 逻辑运算关系表

与			或			非	
Q0.0 = I0.1*I0.1			Q0.1 = I0.2+I0.3			Q0.2 = $\overline{\text{I0.4}}$	
I0.0	I0.1	Q0.0	I0.2	I0.3	Q0.1	I0.4	Q0.2
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

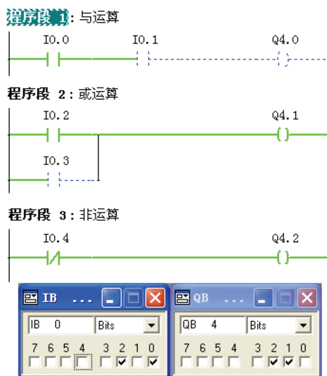


图 3-1 与或非逻辑运算的仿真



2．检验基本逻辑运算的实验

用“新建项目”向导生成一个名为“位逻辑”的项目（见随书光盘中的同名例程），CPU 为 CPU 312C。

打开 OB1，输入图 3-1 所示的梯形图程序，其中的 I、Q 点的地址和它们之间的逻辑关系与表 3-1 中的相同。

打开 PLCSIM，将用户程序和系统数据下载到仿真 PLC，将仿真 PLC 切换到 RUN 模式。按表 3-1 的要求逐行检验“与”、“或”、“非”逻辑运算。以“与”逻辑运算为例，按表格中第一行的要求，令 I0.0 和 I0.1 均为 0 状态（小方框内没有“√”，常开触点断开），观察 Q4.0 是否为 0 状态（线圈断电）；按表格中第二行的要求，令 I0.0 为 0 状态，I0.1 为 1 状态（小方框内有“√”，常开触点闭合），观察 Q4.0 是否为 0 状态……

3．RLO 边沿检测指令

图 3-2 中 I0.5 和 I0.6 的触点组成的串联电路（其逻辑表达式为 $I0.5*\overline{I0.6}$）由断开变为接通时，中间标有“P”的上升沿检测元件左边的逻辑运算结果（RLO）由 0 变为 1（即波形的上升沿），检测到一次正跳变。能流只在该扫描周期内流过检测元件，M0.1 的线圈仅在这一个扫描周期内“通电”。图 3-3 是有关信号的波形图，高电平表示 1 状态，低电平表示 0 状态。M0.1 和 M0.3 的脉冲宽度只有一个扫描周期。

单击工具栏上的按钮，起动程序状态监控功能。流过 M0.1 线圈的脉冲宽度太窄，监控时不一定能看到流过 M0.1 的线圈和触点的能流的快速闪动。在做仿真实验时，需要多次单击 I0.5 对应的小方框，断开然后接通流进上升沿检测元件的能流，才有可能看到它。

边沿检测元件的地址（例如图 3-2 中的 M0.0 和 M0.2）为边沿存储位，用来储存上一次扫描循环的逻辑运算结果。

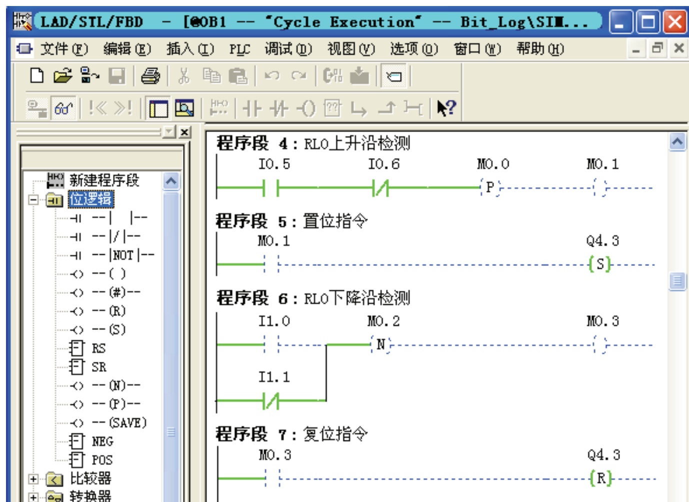


图 3-2 RLO 边沿检测与置位复位指令

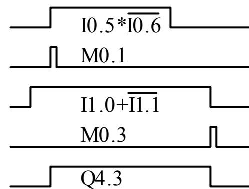


图 3-3 波形图

为了在梯形图中生成常开触点、常闭触点和线圈之外的元件，例如图 3-2 中的上升沿检测元件，单击工具栏上的按钮，在出现的输入框中输入“P”（见图 3-4），或者向下拉动滚动条中的滑块，双击指令列表框中的“P”。

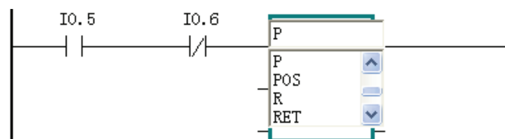


图 3-4　生成上升沿检测元件

也可以用另一种方法生成上升沿检测元件：执行菜单命令“查看”→“总览”，显示出图 3-2 左边的指令列表窗口。打开其中的“位逻辑”文件夹，用鼠标左键单击并按住其中的--(P)--图标，将它“拖”到梯形图中需要放置的地方，光标变为，表示可以在该处放置元件。放开按住的鼠标左键，元件被放置在光标所在的位置。

选中指令列表中的某一条指令，在下面的小窗口可以看到该指令的简要说明。

单击图 3-4 中常闭触点右边的“导线”，该段导线被选中。双击指令列表中的--(P)--图标，它将在常闭触点右边出现。

图 3-2 中 I1.0 和 I1.1 的触点组成的并联电路（其逻辑表达式为 $\mathrm{I1.0}+\overline{\mathrm{I1.1}}$）由接通变为断开时（即图 3-3 中波形的下降沿），中间标有“N”的检测元件左边的逻辑运算结果由 1 变为 0，检测到一次负跳变，能流只在该扫描周期内流过检测元件，M0.3 的线圈仅在这一个扫描周期内“通电”（见图 3-3）。

4．置位指令与复位指令

S (Set，置位或置 1)指令将指定的位地址置位（变为 1 状态并保持）。图 3-2 中 M0.1 的常开触点接通时，Q4.3 变为 1 状态并保持该状态，即使 M0.1 的常开触点断开，它也仍然保持 1 状态。

R (Reset，复位或置 0)指令将指定的地址位复位（变为 0 状态并保持）。图 3-2 中 M0.3 的常开触点闭合时，Q4.3 变为 0 状态并保持该状态。即使 M0.3 的常开触点断开，它也仍然保持 0 状态。

在做仿真实验时，用鼠标单击视图对象 IB0 和 IB1 中对应的小方框，观察在 I0.5 和 I0.6 的触点组成的串联电路由断开到接通（上升沿）时是否能将 Q4.3 置位，在 I1.0 和 I1.1 的触点组成的并联电路由接通变为断开（下降沿）时是否能将 Q4.3 复位。

5．地址边沿检测指令

POS 是单个地址位信号的上升沿检测指令，相当于一个常开触点。如果图 3-5 中输入信号 I1.2 由 0 状态变为 1 状态（即 I1.2 的上升沿），POS 指令等效的常开触点闭合，Q 输出端在一个扫描循环周期内有能流输出，Q4.4 被置位为 1 状态。图中的 M0.4 为边沿存储位，用来储存上一次扫描循环时 I1.2 的状态。

NEG 是单个地址位信号的下降沿检测指令，相当于一个常开触点。如果图 3-5 中的 I1.3 由 1 状态变为 0 状态（即输入信号 I1.3 的下降沿），NEG 指令等效的常开触点闭合，Q 输出端在一个扫描循环周期内有能流输出，Q4.4 被复位为 0 状态。M0.5 为边沿存储位。

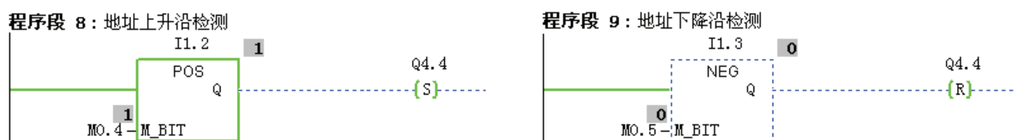


图 3-5　单个位地址位信号的下降沿检测指令



在做仿真实验时，用鼠标单击视图对象 IB1 中 I1.2 或 I1.3 对应的小方框，观察在 I1.2 的上升沿时（小方框内出现“ √ ”）是否能将 Q4.4 置位，在 I1.3 的下降沿（小方框内的“ √ ”消失）是否能将 Q4.4 复位。在 I1.2 的上升沿或 I1.3 的下降沿，用程序状态监控功能可能看到从 POS 或 NEG 方框的 Q 输出端流出的能流快速闪动一下。

6. SR 触发器与 RS 触发器

SR 触发器与 RS 触发器的输入/输出关系见表 3-2，它们的 S 输入为 1、R 输入为 0 时，输出 Q 为 1；S 输入为 0、R 输入为 1 时，输出 Q 为 0。二者的区别在于 S 和 R 输入均为 1 时，SR 触发器的输出 Q 为 0，RS 触发器的输出 Q 为 1（见图 3-6）。

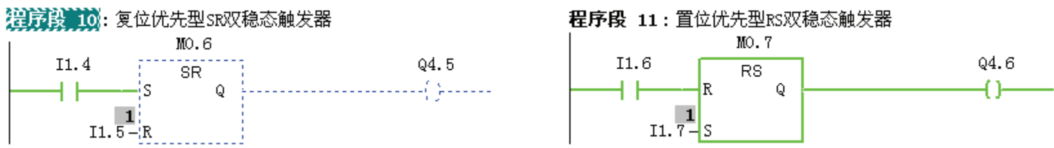


图 3-6　SR 触发器与 RS 触发器

做仿真实验时，按表 3-2 分别改变两种触发器的 S、R 输入信号的状态，观察是否能满足表 3-2 的要求。

7. 取反触点

取反触点的中间标有“NOT”，用来将它左边电路的逻辑运算结果（RLO）取反（见图 3-7），该运算结果若为 1 则变为 0，若为 0 则变为 1。

做仿真实验时，可以看到 I2.0 和 I2.1 的触点组成的串联电路接通时，有能流到达取反触点，但是该触点没有能流输出。I2.0 和 I2.1 的触点组成的串联电路断开时，没有能流流进取反触点，但是该触点有能流输出，Q 4.7 的线圈通电（见图 3-7）。

表 3-2　输入输出关系表

SR 触发器			RS 触发器		
S	R	Q	S	R	Q
0	0	不变	0	0	不变
0	1	0	0	1	0
1	0	1	1	0	1
1	1	0	1	1	1



图 3-7　能流取反

8. 中间输出

中间输出（又称为中线输出）是一种中间分配单元，用该元件指定的地址来保存它左边电路的逻辑运算结果（RLO）。中间标有“#”号的中间输出线圈与其他触点串联（见图 3-8），它并不影响能流的流动。中间输出只能放在梯形图的中间，不能接在左侧的垂直“电源线”上，也不能放在最右端电路结束的位置。

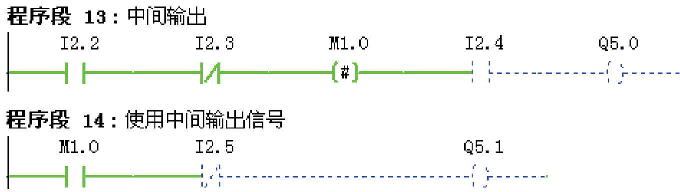


图 3-8　中间输出指令

做仿真实验时，接通 I2.2 和 I2.3 的触点组成的串联电路，中间输出线圈通电。因为它对应的 M1.0 变为 1 状态，程序段 14 中 M1.0 的常开触点闭合。断开 I2.2 和 I2.3 的触点组成的串联电路，中间输出线圈断电，M1.0 的常开触点断开。

9. 仿真练习

在地下停车场的出入口处，同时只允许一辆车进出，在进出通道的两端设置有红绿灯（见图 3-9），光电开关 I0.0 和 I0.1 用来检测是否有车经过，光线被车遮住时，I0.0 或 I0.1 为 1 状态。有车进入通道时（光电开关检测到车的前沿），两端的绿灯灭，红灯亮，以警示两方后来的车辆不能再进入通道。车离开通道时，光电开关检测到车的后沿，两端的红灯灭，绿灯亮，别的车辆可以进入通道。

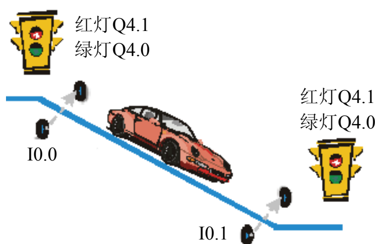


图 3-9　停车场入口示意图

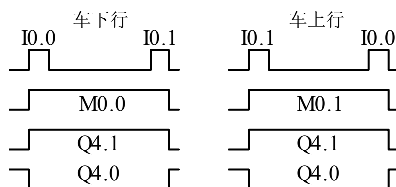


图 3-10　信号波形图

图 3-10 的波形图中的 M0.0 和 M0.1 分别是有车下行和有车上行的信号。

用“新建项目”向导生成一个名为“车库入口”的项目（见随书光盘中的同名例程），CPU 可选任意的型号。

用符号表定义符号地址（见图 3-11）。将图 3-12 中的梯形图程序输入 OB1。

符号编辑器 - [S7 程序(1) (符号) -- 车库入口\SIM...

符号表(S)　编辑(E)　插入(I)　视图(V)　选项(O)　窗口(W)　帮助(H)

全部符号

	状态	符号	地址	数据类型	注释
1		上入口	I 0.0	BOOL	
2		下入口	I 0.1	BOOL	
3		车下行	M 0.0	BOOL	
4		车上行	M 0.1	BOOL	
5		绿灯	Q 4.0	BOOL	
6		红灯	Q 4.1	BOOL	

图 3-11　符号表

打开 PLCSIM，将用户程序和系统数据下载到仿真 PLC。将仿真 PLC 切换到 RUN-P 模式。在 PLCSIM 中生成 IB0、QB4 和 MB0 的视图对象。

按波形图分别模拟有车下行和有车上行，观察 M0.0、M0.1 和信号灯的状态变化是否满足要求。

1）两次单击 I0.0 对应的小方框，模拟有车下行经过上限位开关。观察 M0.0、Q4.1 和 Q4.0 的状态变化。

2）两次单击 I0.1 对应的小方框，模拟有车下行经过下限位开关。观察 M0.0、Q4.1 和 Q4.0 的状态变化。



3）用同样的方法，模拟有车上行的情况。

图 3-12 中的常闭触点有什么作用？短接这两个触点后下载 OB1，模拟有车下行和有车下行，通过观察到的现象分析常闭触点的作用。

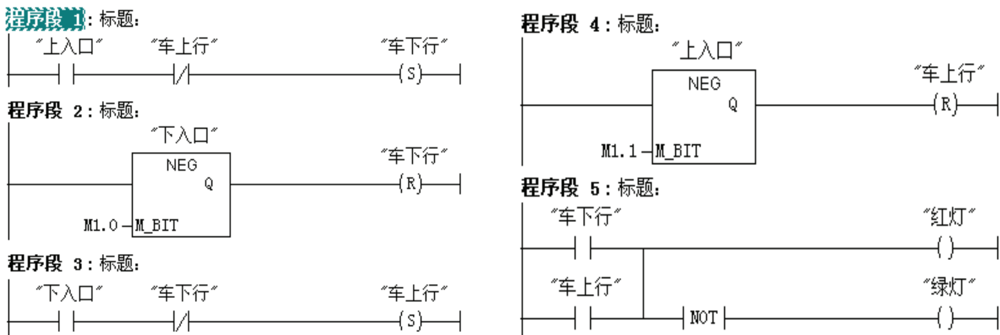


图 3-12　梯形图

3.1.2　实训六　故障显示电路

1．实验要求

故障信号 I2.6 为 1 状态时，Q5.2 控制的指示灯以 1Hz 的频率闪烁。操作人员按复位按钮 I2.7 后，如果故障已经消失，指示灯熄灭。如果没有消失，指示灯转为常亮，直至故障消失（见图 3-13）。

2．实验步骤

（1）将图 3-13 所示的故障信息显示电路输入到 OB1，也可以打开随书光盘中的项目“位逻辑”

故障信号 I2.6 的上升沿用 POS 指令检测，它输出的一个扫描循环周期的脉冲作为起动保持停止电路的起动电路，使 M1.3 为 1 并保持。即使在按下停止按钮 I2.7 时故障信号尚未消失，也能使 M1.3 变为 0 状态（见图 3-14）。

（2）设置时钟存储器字节

指示灯的闪烁用时钟存储器位 M4.5 来实现。所谓的时钟是周期恒定、占空比为 0.5（脉冲宽度与周期之比为 50%）的脉冲信号。S7-300/400 有一个需要设置地址的时钟存储器字节，该字节的 8 位提供 8 个不同周期的时钟脉冲。

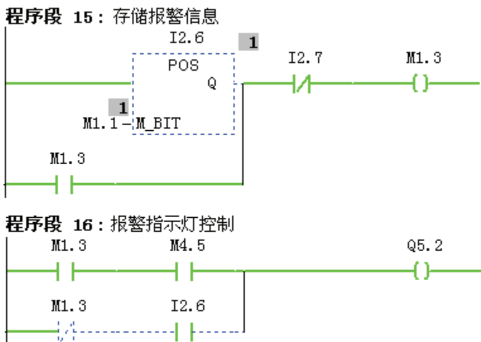


图 3-13　故障信息显示电路

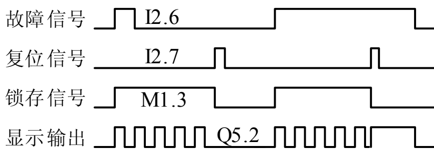


图 3-14　波形图

双击 HW Config 的机架中 CPU 模块所在的行，打开 CPU 的“属性”对话框的“周期/时钟存储器”选项卡（见图 3-15）。选中“时钟存储器”复选框，设置时钟存储器（M）的字节地址为 4，即 MB4 为时钟存储器字节。

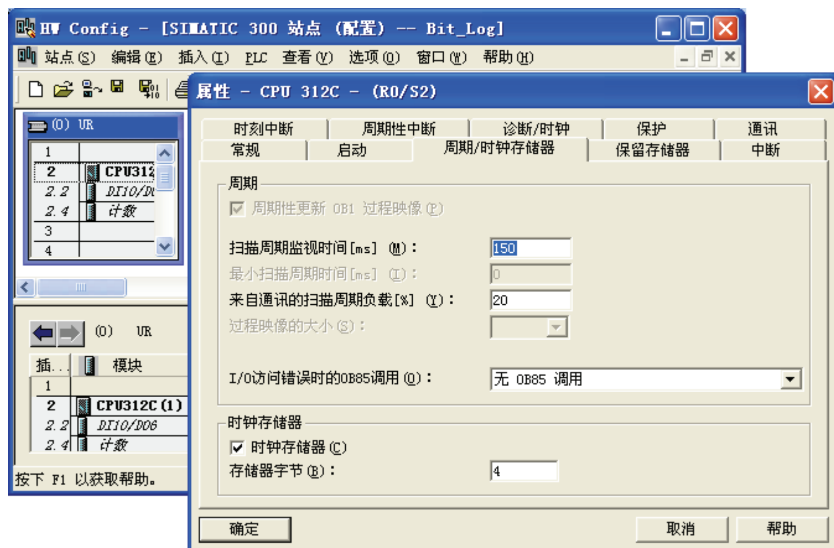


图 3-15　设置时钟存储器字节

（3）使用在线帮助功能

用下面的方法可以查到时钟存储器字节内各位的时钟脉冲的周期或频率：

打开 CPU 的“属性”对话框的“周期/时钟存储器”选项卡之后，按计算机的〈F1〉键，或单击选项卡中的“帮助”按钮，打开该选项卡的在线帮助信息。帮助文件中的绿色字符链接有更多的帮助信息。

单击绿色的“时钟存储器”，可以看到有关它的信息：“时钟存储器是周期性改变其二进制数值的存储位(脉冲占空比：1:1)。当要使用时钟存储器时激活该复选框，然后输入存储字节的数目。”单击其中绿色的“周期性”，可以看到表 3-3 中的时钟存储器各位的时钟脉冲周期与频率，其中的第 5 位（本例中为 M4.5）的周期为 1s。

表 3-3　时钟存储器字节各位对应的时钟脉冲周期与频率

位	7	6	5	4	3	2	1	0
周期/s	2	1.6	1	0.8	0.5	0.4	0.2	0.1
频率/Hz	0.5	0.625	1	1.25	2	2.5	5	10

（4）仿真实验

1）打开 PLCSIM，将用户程序和系统数据下载到仿真 PLC。将仿真 PLC 切换到 RUN-P 模式。打开 OB1，单击工具栏上的按钮，进入程序状态监控（见图 3-13）。可以看到梯形图中 M4.5 的常开触点以 1Hz 的频率不断变化。

2）单击 PLCSIM 中 I2.6 对应的小方框，模拟故障信号出现。梯形图中 M1.3 的线圈通电，M1.3 的常开触点接通，Q5.2 的线圈以 1Hz 的频率反复接通和断开，Q5.2 控制的指示灯闪烁。再次单击 I2.6 对应的小方框，方框中的“√”消失，I2.6 变为 0 状态，模拟故障消



失，指示灯继续闪烁。

3）连续单击两次 I2.7 对应的小方框，模拟操作人员按下和松开复位按钮，M1.3 的线圈断电，其常开触点断开，Q5.2 的线圈断电，指示灯停止闪烁。

4）单击 PLCSIM 中 I2.6 对应的小方框，M1.3 的线圈通电，指示灯闪烁。

5）连续单击两次 I2.7 对应的小方框，模拟在故障信号未消失时操作人员按下和松开复位按钮。M1.3 的线圈断电，其常开触点断开，图 3-13 中程序段 16 上面的串联电路断开。M1.3 的常闭触点闭合，因为此时故障信号 I2.6 为 1，程序段 16 下面的串联电路接通，指示灯由闪烁变为常亮。

6）单击 I2.6 对应的小方框，使 I2.6 变为 0 状态，它的常开触点断开，故障消失，程序段 16 下面的串联电路断开，Q5.2 的线圈断电，指示灯熄灭。

3.2 定时器计数器指令应用

3.2.1 实训七 定时器指令的基本功能

S7-300/400 有 5 种定时器，每种定时器在梯形图中又有两种表示方法，S5 定时器用指令框（Box）的形式来表示，此外每一种 S5 定时器都有功能相同的定时器线圈。

1．定时器的种类和存储区

S7-300/400 的定时器分为脉冲定时器、扩展的脉冲定时器、接通延时定时器、保持型接通延时定时器和断开延时定时器。各种定时器的输入/输出基本功能如图 3-16 所示，其中的“*t*”是定时器的时间预置值。

每个定时器有一个 16 位的字和一个二进制的位，定时器的字用来存放它的剩余时间值，定时器触点的状态由它的位的状态来决定。S7-300 的定时器个数（128～2048 个）与 CPU 的型号有关，S7-400 的 CPU 有 2048 个定时器。

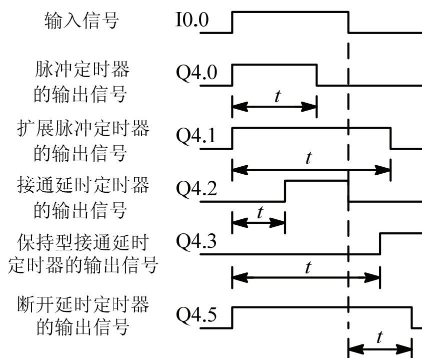


图 3-16 定时器功能

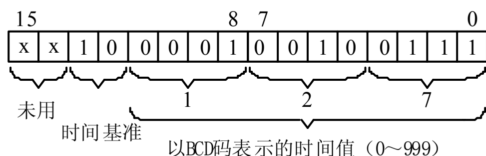


图 3-17 定时器字

2．定时器字的表示方法

用户使用的定时器字由 3 位 BCD 码时间值（0～999）和时间基准组成（见图 3-17）。

定时器字的第 12 位和第 13 位用来作时间基准，二进制数 00、01、10 和 11 对应的时间基准分别为 10ms、100ms、1s 和 10s。实际的定时时间等于时间值乘以时间基准值。例如定

时器字为 W#16#2127 时（见图 3-17），时间基准为 1s，定时时间为 127×1=127s。时间基准越小，分辨率越高，可定时的时间越短；时间基准越大，分辨率越低，可定时的时间越长。可定时的最大时间值为 9990s。

3．定时器预置值的表示方法

在梯形图中使用“S5T#aHbMcSdMS”格式的时间值，a、b、c、d 分别是小时、分、秒和毫秒的值。也可以以秒为单位输入，例如输入时间预置值 S5T#100S 后按回车键，将自动转换为 S5T#1M40S。在语句表中，可以使用 IEC 格式的时间值，例如 T#10S。

4．使用指令的在线帮助

用“新建项目”向导生成一个名为“定时器 1”的项目（见随书光盘中的同名例程），CPU 为 CPU 312C。打开 OB1，选择编程语言为梯形图。打开左边的指令列表窗口中的“定时器”文件夹，选中 S_PULSE(S5 脉冲定时器)，下面的小窗口中是该指令的简要说明（见图 3-18）。按计算机的〈F1〉键，出现该指令的在线帮助文件。

帮助文件中有指令的输入/输出参数的数据类型、允许使用的存储区和参数的意义（见表 3-4）。表中的 L 是程序块中的局部数据区的缩写，D 是数据块的缩写。其他 S5 定时器的输入/输出参数与 S5 脉冲定时器的相同。此外帮助文件中还有对指令的描述、定时器的时序图、指令的执行对状态字的影响，以及指令应用的实例。

读者在学习指令时，重点应放在了解指令的功能上，可以通过在线帮助来了解指令应用中的细节问题，没有必要死记这些细节。有的指令很少使用，不熟悉也没有关系，在读程序遇到它们时，可以通过指令的在线帮助来了解它们。

表 3-4 S5 脉冲定时器的输入/输出参数

参数	数据类型	存储区	描述
S	BOOL	I、Q、M、L、D	使能输入
TV	S5TIME	I、Q、M、L、D 和常数	时间预置值
R	BOOL	I、Q、M、L、D	复位输入
BI	WORD	I、Q、M、L、D	剩余时间值，不带时间基准的十六进制整型格式
BCD	WORD	I、Q、M、L、D	剩余时间值，BCD 格式（S5T#格式）
Q	BOOL	I、Q、M、L、D	定时器的状态位

剩余时间值为时间预置值 TV 减去定时器启动后经过的时间。项目“定时器 1”的 OB1 中有 5 种 S5 定时器的定时电路。

5．S5 脉冲定时器的仿真实验

脉冲定时器类似于数字电路中上升沿触发的单稳态电路，将图 3-18 左边窗口中的 S5 脉冲定时器（S_PULSE）直接“拖放”到梯形图中适当的位置。可以不给 BI 和 BCD 输出端指定地址。

可以根据定时器的时序图来做仿真实验，以此来理解定时器的功能，仿真步骤如下：

1）打开 PLCSIM，将 OB1 和系统数据下载到仿真 PLC。将仿真 PLC 切换到 RUN 或 RUN-P 模式。



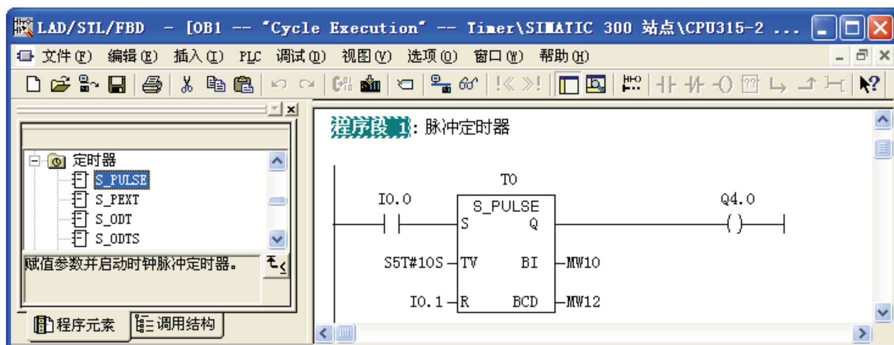


图 3-18　S5 脉冲定时器

2）打开 OB1，单击工具栏上的按钮，起动程序状态监控功能（见图 3-19）。

3）脉冲定时器从输入信号 I0.0 的上升沿开始，输出一个脉冲信号。令输入脉冲的宽度大于等于时间预置值 10s（见图 3-20 中 I0.0 的脉冲 A），Q4.0 输出的脉冲宽度等于 T0 的时间预置值 *t*。

单击 PLCSIM 窗口中 I0.0 对应的小方框，方框内出现“√”。由于输入电路（I0.0 的常开触点）闭合，梯形图中的触点、方框和 Q4.0 的线圈均变为绿色（见图 3-19），表示 T0 正在输出脉冲。T0 被启动后，从预置值开始，每经过一个时间基准，它的剩余时间值减 1。剩余时间值减为 0 时，定时时间到，Q4.0 的线圈断电。在定时期间，BI 端输出十六进制的剩余时间值，BCD 端输出 S5T#格式的剩余时间值。图 3-20 中的时序图用下降的斜坡表示定时期间剩余时间值递减，图中的 *t* 是定时器的时间预置值。

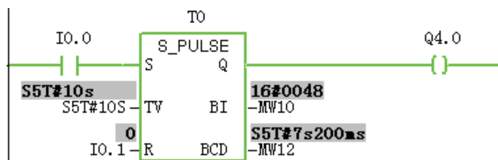


图 3-19　S5 脉冲定时器的程序状态监控

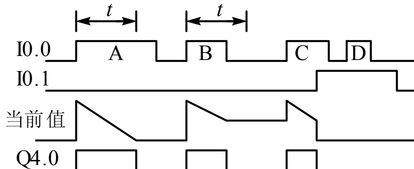


图 3-20　S5 脉冲定时器时序图

4）令 I0.0 为 0 状态，再将它置为 1 状态，起动定时。未到时间预置值 10s 时令 I0.0 变为 0 状态（见图 3-20 中 I0.0 的脉冲 B），Q4.0 的线圈同时断电，剩余时间值保持不变，Q4.0 输出的脉冲宽度等于 I0.0 的输入脉冲的宽度。

5）在 I0.0 下一个上升沿，从时间预置值开始定时。在定时期间令复位信号 I0.1 为 1 状态，定时器被复位（见 I0.0 的脉冲 C）。复位后定时器的剩余时间值被清零，Q4.0 变为 0 状态。

复位信号总是优先的，与其他输入信号的状态无关。在复位信号为 1 时，即使有输入信号出现（见 I0.0 的脉冲 D），Q4.0 也不能输出脉冲。

各种定时器的复位信号的功能相同。

6. S5 扩展脉冲定时器

S5 扩展脉冲定时器(见图 3-21)的功能与脉冲定时器基本上相同，其区别在于前者在输入脉冲宽度小于时间预置值时，也能输出设定宽度的脉冲。仿真实验的步骤如下：

1）令 I0.2 为 1 状态，其常开触点由断开变为接通，T1 开始定时，Q4.1 的线圈通电。定

时时间到时，Q4.1 的线圈断电（见图 3-22 中的波形 A）。

2）令 I0.2 为 0 状态，再将它置为 1 状态，T1 开始定时，定时时间未到时使 I0.2 变为 0 状态，T1 仍继续定时（见 I0.2 的波形 B 和 C）。

3）在 I0.2 的波形 B 的上升沿起动 T1 定时，定时期间令 I0.2 为 0 状态，然后变为 1 状态。在 I0.2 的波形 C 的上升沿，T2 被重新启动，从时间预置值开始定时，直到定时时间到。

4）令 I0.2 为 0 状态，再将它置为 1 状态，T1 开始定时。在定时期间令复位信号 I0.3 为 1 状态，在 I0.3 波形 D 的上升沿，T1 被复位，它的剩余时间值被清零，Q4.1 的线圈断电。

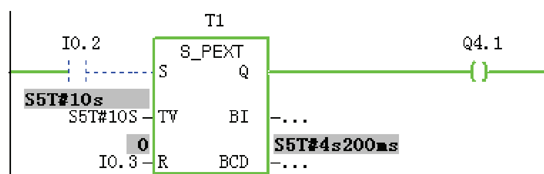


图 3-21　S5 扩展脉冲定时器

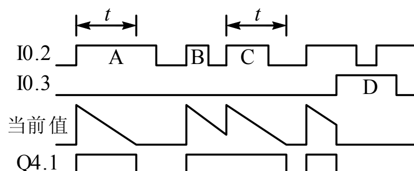


图 3-22　时序图

7. S5 接通延时定时器

接通延时定时器是使用得最多的定时器。S5 接通延时定时器的仿真实验的步骤如下：

1）令 I0.4 为 1 状态（见图 3-23），其常开触点使 T2 的输入电路接通，开始定时，T2 的剩余时间值不断减 1。减至 0 时，定时时间到，其 Q 输出端变为 1 状态，Q4.2 的线圈通电（见图 3-24 中 I0.4 的波形 A）。

2）令 I0.4 变为 0 状态，在 I0.4 的波形 A 的下降沿，I0.4 的常开触点断开，Q4.2 的线圈断电。

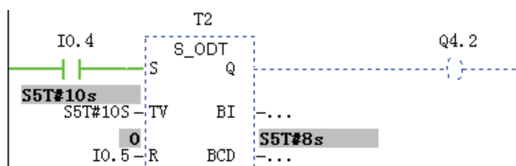


图 3-23　S5 接通延时定时器

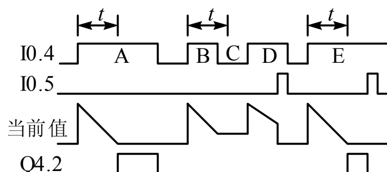


图 3-24　时序图

3）令 I0.4 为 1 状态，在它的波形 B 的上升沿，T1 开始定时。定时时间未到时使 I0.4 变为 0 状态（见 I0.4 的波形 C），T2 的剩余时间值保持不变。

4）在 I0.4 的波形 D 的上升沿，T2 从时间预置值开始定时。用 I0.5 产生一个复位脉冲，T2 被复位，它的剩余时间值被清零。

5）令 I0.4 为 0 状态，再将它置为 1 状态，在 I0.4 的波形 E 的上升沿，T2 开始定时。定时结束后 Q4.2 的线圈通电。在 I0.5 第 2 个复位脉冲的上升沿，Q4.2 的线圈断电。

综上所述，可以用断开输入电路和使用复位信号这两种方法，使接通延时定时器的输出位 Q 变为 0 状态。

8. S5 保持型接通延时定时器

保持型接通延时定时器(见图 3-25)的功能与接通延时定时器的基本相同，其区别在于前



者在输入脉冲宽度小于时间预置值时，也能正常定时。S5 保持型接通延时定时器的仿真实验步骤如下：

1）令 I0.6 为 1 状态，其常开触点由断开变为接通，T3 开始定时，它的剩余时间值不断减 1。定时时间未到时，令 I0.6 为 0 状态（见图 3-26 中 I0.6 的波形 A），T3 继续定时。定时时间到时，Q4.3 的线圈通电。

2）令复位信号 I0.7 为 1（见 I0.7 的波形 E），T3 被复位，Q4.3 的线圈断电。只能用复位信号使保持型通电延时定时器复位。

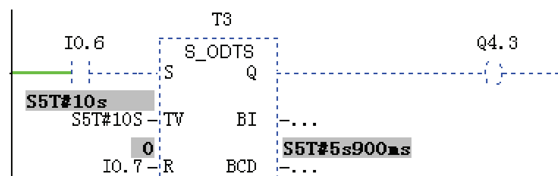


图 3-25　S5 保持型接通延时定时器

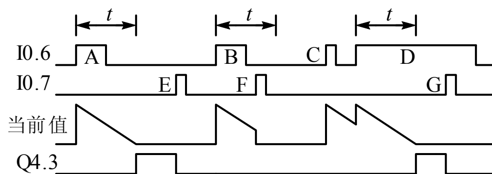


图 3-26　时序图

3）令 I0.6 为 1 状态后又将它置为 0 状态（见 I0.6 的波形 B），用脉冲起动 T3 开始定时。在定时期间令 T3 的复位信号 I0.7 为 1（见 I0.7 的波形 F），T3 被复位，其剩余时间值被清零。

4）用 I0.6 的窄脉冲起动 T3 定时（见 I0.6 的波形 C），定时时间未到时，再次令 I0.6 为 1 状态（见 I0.6 的波形 D），T3 又从时间预置值开始定时。

5）在输入信号 I0.6 为 1 时，复位信号（见 I0.7 的波形 G）也能使 T3 复位。

9．S5 断开延时定时器

某些主设备（例如大型变频调速电动机）在运行时需要用风扇冷却，停机后风扇应延时一段时间才能断电。可以用断开延时定时器(见图 3-27)来方便地实现这一功能，即用反映主设备运行的信号 I1.0 作为断开延时定时器的输入信号 S，用断开延时定时器的输出 Q 控制风扇 Q4.4。仿真实验的步骤如下：

1）令 I1.0 为 1 状态，其常开触点接通，T4 的输出位变为 1 状态（见图 3-28），Q4.4 的线圈通电。

2）令 I1.0 为 0 状态，T4 开始定时。其剩余时间值减为 0 时，定时时间到，Q4.4 的线圈才断电。

3）在起动信号 I1.0 波形 B 的下降沿，T4 开始定时。在定时期间，复位脉冲 I1.1 将 T4 复位，T4 的剩余时间值被清零，Q4.4 的线圈断电。

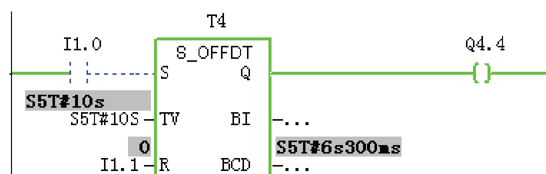


图 3-27　S5 断开延时定时器

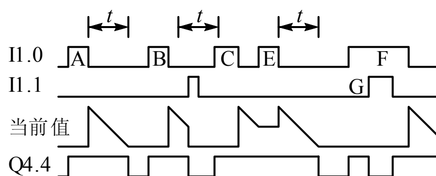


图 3-28　时序图

4）在起动信号 I1.0 波形 C 的下降沿，T4 开始定时。在定时期间，令 I1.0 为 1 状态（见 I1.0 的波形 E），T4 的剩余时间值保持不变。在波形 E 的下降沿，T4 又从时间预置值开始定时。

5）在输入信号 I1.0 为 1 时（见 I1.0 的波形 F），复位信号（见 I1.1 的波形 G）也能使 T4 复位。

10．脉冲定时器线圈指令的仿真实验

用“新建项目”向导生成一个名为“定时器 2”的项目（见随书光盘中的同名例程），CPU 可以采用任意的型号。在 OB1 中生成使用 5 种定时器线圈指令的定时电路。

图 3-29 中的脉冲定时器线圈电路与图 3-18 中的 S5 脉冲定时器的功能、输入/输出位地址和时序图相同，仿真的步骤也完全相同。

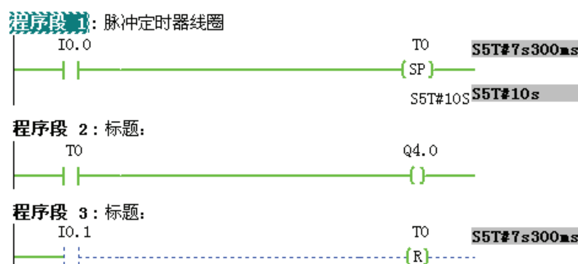


图 3-29　脉冲定时器线圈

脉冲定时器线圈电路的仿真步骤如下：

1）将 OB1 和系统数据下载到仿真 PLC，将仿真 PLC 切换到 RUN 或 RUN-P 模式。

2）打开 OB1，单击工具栏上的按钮，起动程序状态监控功能。

3）令 I0.0 为 1 状态，其常开触点由断开变为接通，T0 开始定时，其常开触点闭合。它的剩余时间值减至 0 时，定时时间到，T0 的常开触点断开。

4）令 I0.0 为 0 状态，再将它置为 1 状态，T0 开始定时，定时时间未到时使 I0.0 变为 0 状态，T0 的常开触点断开，剩余时间值保持不变。

5）令 I0.0 为 0 状态，再将它置为 1 状态，T0 开始定时。在定时期间令复位信号 I0.1 为 1 状态，T0 被复位，它的剩余时间值被清零，其常开触点断开。

其他 4 种定时器线圈的定时电路（见随书光盘中的项目“定时器 2”）与图 3-29 中的类似。它们的工作过程、输入/输出位地址、时序图和仿真实验的方法，与对应的 S5 定时器的相同。

3.2.2　实训八　卫生间冲水控制电路

S7-300/400 的定时器种类较多，巧妙地应用各种定时器，可以简化电路，方便地实现较为复杂的控制功能。本实训的卫生间冲水控制电路使用了 3 种定时器。

1．程序设计

I1.2 是光电开关检测到的有使用者的信号（见图 3-30），用 Q4.5 控制冲水电磁阀。从 I1.2 的上升沿（有人使用）

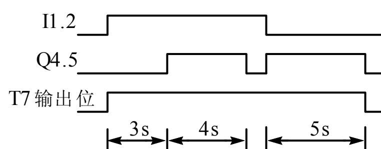


图 3-30　波形图



开始，用接通延时定时器 T5 实现 3s 的延时，3s 后 T5 的常开触点接通，使脉冲定时器 T6 的线圈通电，T6 的常开触点输出一个 4s 的脉冲。从 I1.2 的上升沿开始，断开延时定时器 T7 的常开触点接通。使用者离开时（在 I1.2 的下降沿），断开延时定时器开始定时，5s 后 T7 的常开触点断开，停止冲水。

由波形图可知，控制冲水电磁阀的 Q4.5 输出的高电平脉冲波形由两块组成，4s 的脉冲波形由脉冲定时器 T6 的常开触点提供。T7 输出位的波形减去 I1.2 的波形得到宽度为 5s 的脉冲波形，可以用 T7 的常开触点与 I1.2 的常闭触点串联的电路来实现上述要求。两块脉冲波形的叠加用并联电路来实现。

2．实验步骤

用“新建项目”向导生成一个项目，可以选用任意型号的 CPU，将图 3-31 中的程序输入 OB1。随书光盘中的例程“定时器 1”也有图 3-31 中的程序。仿真实验的步骤如下：

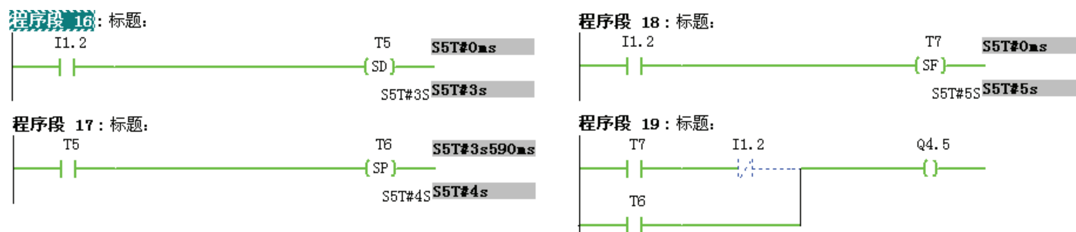


图 3-31　卫生间冲水控制电路

1）将 OB1 和系统数据下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。

2）打开 OB1，单击工具栏上的按钮，起动程序状态监控功能。

3）单击 PLCSIM 中 I1.2 对应的小方框，方框中出现“√”，I1.2 变为 1 状态。从被监控的梯形图可以看出，接通延时定时器 T5 开始定时，同时断开延时定时器 T7 的常开触点接通。经过 3s 的延时后，T5 的常开触点闭合，脉冲定时器 T6 开始定时，其常开触点闭合，使 Q4.5 的线圈通电，电磁阀开始冲水。4s 后 T6 的常开触点断开，Q4.5 的线圈断电，停止冲水。

4）单击 PLCSIM 中 I1.2 对应的小方框，方框中的“√”消失，I1.2 变为 0 状态。此时 T7 开始定时，T7 的常开触点和 I1.2 的常闭触点组成的串联电路接通，Q4.5 的线圈通电，电磁阀冲水。5s 后 T7 的常开触点断开，Q4.5 的线圈断电，停止冲水。

3．编程与仿真练习

按下起动按钮 I0.0，Q4.0 控制的电机运行 20s，然后自动断电，同时 Q4.1 控制的制动电磁铁开始通电，10s 后自动断电。用扩展的脉冲定时器和断开延时定时器设计控制电路。

用“新建项目”向导生成一个项目，CPU 可选任意的型号。在 OB1 中编写满足上述要求的程序，读者自己拟定仿真实验的步骤，检验程序运行的结果。

3.2.3　实训九　运输带控制系统

1．两条运输带的控制实验

两条运输带顺序相连(见图 3-32)，为了避免运送的物料在 1 号运输带上堆积，按下起动按钮 I0.0，1 号运输带开始运行，8s 后 2 号运输带自动起动（见图 3-34）。停机的顺序与起动的顺序刚好相反，即按了停止按钮 I0.1 后，先停 2 号运输带，8s 后停 1 号运输带。PLC

通过 Q4.0 和 Q4.1 控制两台电动机 M1 和 M2。图 3-33 是 PLC 的外部接线图。

用“新建项目”向导生成一个名为“运输带控制”的项目（见随书光盘中的同名例程），CPU 可以采用任意的型号。

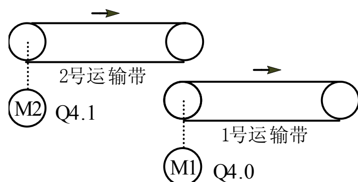


图 3-32　运输带示意图

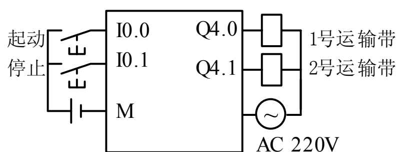


图 3-33　PLC 外部接线图

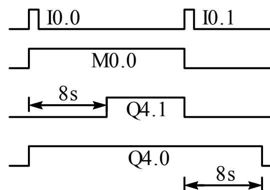


图 3-34　波形图

梯形图程序如图 3-35 所示，程序中设置了一个用起动按钮和停止按钮控制的辅助元件 M0.0，用它的常开触点控制接通延时定时器 T0 和断开延时定时器 T1 的线圈。

接通延时定时器 T0 的常开触点在 I0.0 的上升沿之后 8s 接通，在它的线圈断电（M0.0 的下降沿）时断开。综上所述，可以用 T0 的常开触点直接控制 2 号运输带 Q4.1。

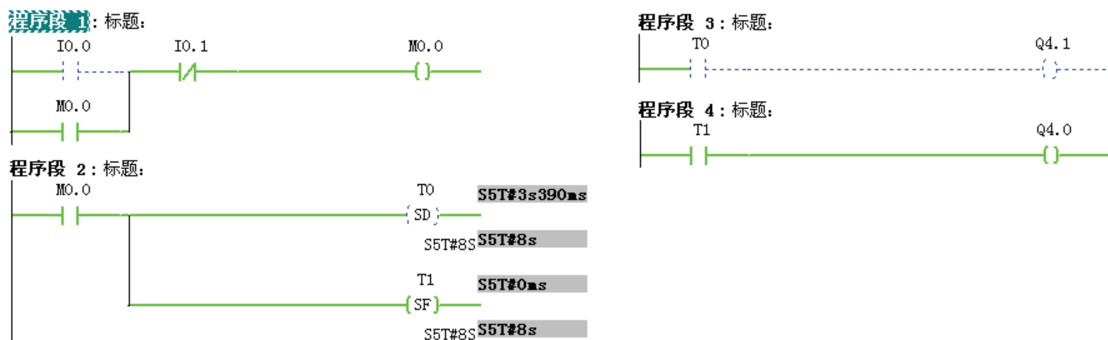


图 3-35　梯形图

断开延时定时器 T2 的常开触点在它的线圈通电时接通，在它结束 8s 延时后断开，因此可以用 T1 的常开触点直接控制 1 号运输带 Q4.0。实验步骤如下：

1）将 OB1 下载到仿真 PLC，将仿真 PLC 切换到 RUN 或 RUN-P 模式。

2）打开 OB1，单击工具栏上的按钮，起动程序状态监控功能。

3）两次单击 PLCSIM 中 I0.0 对应的小方框，模拟按下和放开起动按钮。观察梯形图中 M0.0 和 Q4.0 的线圈是否同时通电，T0 是否开始定时，8s 后 Q4.1 是否能自动变为 1 状态。

4）两次单击 PLCSIM 中 I0.1 对应的小方框，模拟按下和放开停止按钮。观察梯形图中 M0.0 和 Q4.1 的线圈是否同时断电，T1 是否开始定时，8s 后 Q4.0 的线圈是否能自动断电。

2．3 条运输带控制的仿真实验

3 条运输带顺序相连(见图 3-36)，为了避免运送的物料在 1 号和 2 号运输带上堆积，按下起动按钮 I0.2，1 号运输带开始运行，5s 后 2 号运输带自动起动，再过 5s 后 3 号运输带自动起动。停机的顺序与起动的顺序刚好相反，即按了停止按钮 I0.3 后，3 号运输带停机，5s 后 2 号运输带停机，再过 5s 停 1 号运输带。PLC 通过 Q4.2～Q4.4 控制 3 台运输带的电动机 M1、M2 和 M3。



图 3-37 中的波形图给出了程序设计的思路。程序中设置了一个用起动按钮和停止按钮控制的辅助元件 M0.1，用它的常开触点控制接通延时定时器 T2 和断开延时定时器 T5 的线圈。接通延时定时器 T2 的常开触点在 I0.2 的上升沿之后 5s 接通，用它的常开触点控制接通延时定时器 T3 和断开延时定时器 T4 的线圈。

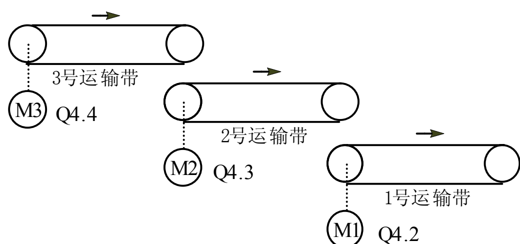


图 3-36　运输带示意图

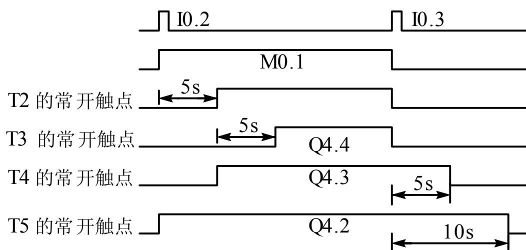


图 3-37　波形图

按下停止按钮，M0.1 变为 0 状态，使 T4 和 T5 的线圈断开，它们开始定时。5s 后 T4 的常开触点断开，10s 后 T5 的常开触点断开。

由波形图可知，可以用 T3、T4 和 T5 的常开触点直接控制 Q4.4、Q4.3 和 Q4.2 的线圈。

根据图 3-37 中的波形图，设计出控制 3 条运输带的梯形图（见图 3-38）。

仿真实验的步骤如下：

1）图 3-38 中的程序在项目“运输带控制”的 OB1 中，将 OB1 和系统数据下载到仿真 PLC，将仿真 PLC 切换到 RUN 或 RUN-P 模式。

2）打开 OB1，单击工具栏上的按钮，起动程序状态监控功能。

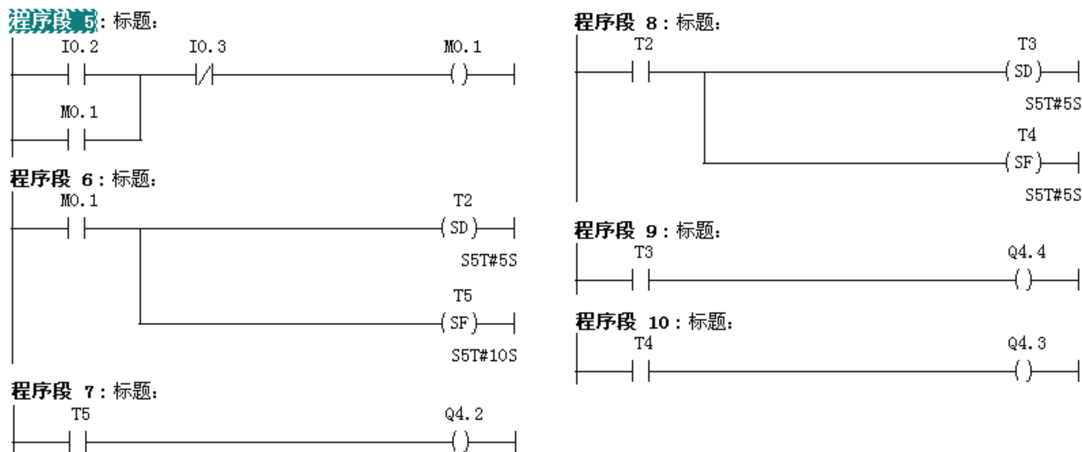


图 3-38　梯形图

3）两次单击 PLCSIM 中 I0.2 对应的小方框，方框中的“√”出现后又消失，以此来模拟按下和松开起动按钮。观察梯形图中 M0.1 和 Q4.2 的线圈是否同时通电，T2 是否开始定时；5s 后 Q4.3 是否能自动变为 1 状态，T3 是否开始定时；再过 5s 后 Q4.4 的线圈是否自动通电。

4）两次单击 PLCSIM 中 I0.3 对应的小方框，模拟按下和放开停止按钮。观察梯形图中 M0.1 和 Q4.4 的线圈是否同时断电，断开延时定时器 T4、T5 是否开始定时，5s 后 Q4.3 是否能自动变为 0 状态，再过 5s 后 Q4.2 是否能自动变为 0 状态。

为了简化调试过程，也可以只观察系统的输入、输出关系。用鼠标产生起动按钮和停止按钮信号后，观察各输出点的状态变化是否符合波形图的要求。如果不符合，再观察定时器的工作是否正常。

3.2.4 实训十　小车控制系统

1．实验要求

图 3-39 中的小车开始时停在左边，左限位开关 SQ1 的常开触点闭合。要求按下列顺序控制小车：

1）按下右行起动按钮 SB2，小车右行。

2）小车走到右限位开关 SQ2 处停止运动，延时 8s 后开始左行。

3）小车回到左限位开关 SQ1 处时停止运动。

2．程序设计

用“新建项目”向导生成一个名为“小车控制 2”的项目（见随书光盘中的同名例程），CPU 可以采用任意的型号。在异步电动机正反转控制电路的基础上设计的满足上述要求的梯形图如图 3-40 所示。在控制右行的 Q4.0 的线圈回路中串联了 I0.4 的常闭触点，小车走到右限位开关 SQ2 处时，I0.4 的常闭触点断开，使 Q4.0 的线圈断电，小车停止右行。同时 I0.4 的常开触点闭合，T6 的线圈通电，开始定时。8s 后定时时间到，T6 的常开触点闭合，使 Q4.1 的线圈通电并自保持，小车开始左行。离开限位开关 SQ2 后，I0.4 的常开触点断开，T6 因为线圈断电，其常开触点断开。小车运行到左边的起始点时，左限位开关 SQ1 动作，I0.3 的常闭触点断开，使 Q4.1 的线圈断电，小车停止运动。

在梯形图中，保留了左行起动按钮 I0.1 和停止按钮 I0.2 的触点，使系统有手动操作的功能。在手动时，起保停电路中的左限位开关 I0.3 和右限位开关 I0.4 的常闭触点可以防止小车的运动超限。

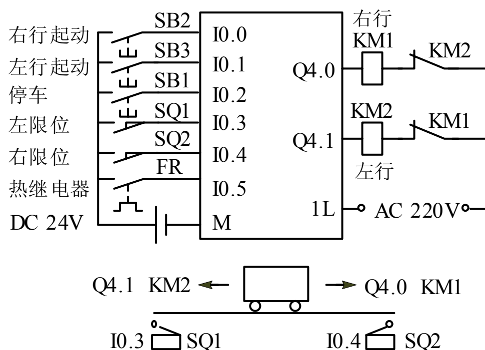


图 3-39　PLC 外部接线图

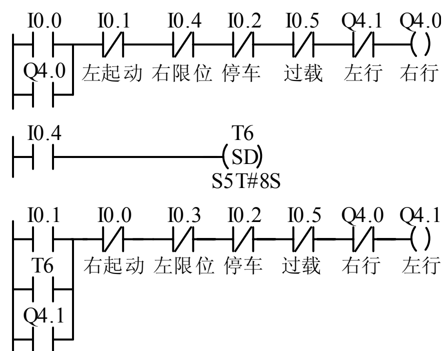


图 3-40　梯形图

3．仿真实验过程

1）将 OB1 和系统数据下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。



2）生成 IB0、QB4 和 T6 对应的视图对象（见图 3-41）。

3）两次单击 PLCSIM 中 I0.0 对应的小方框，方框中的“√”出现后又消失，以此来模拟按下和放开起动按钮。观察控制右行的 Q4.0 是否为 1 状态。

4）单击 I0.4 对应的小方框，方框中出现“√”，模拟右限位开关动作，观察 Q4.0 是否变为 0 状态，小车停止右行；以及接通延时定时器 T6 是否开始定时。

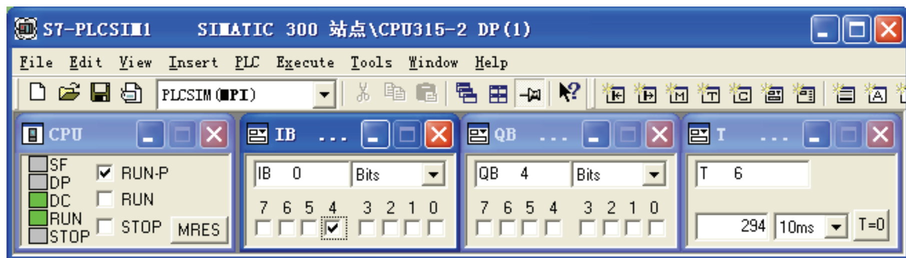


图 3-41　PLCSIM

5）T6 的定时时间到时，观察 Q4.1 是否变为 1 状态，小车左行。

6）小车左行后离开右限位开关 I0.4，应及时将 I0.4 置为 0 状态。

7）令左限位开关 I0.3 为 1 状态，模拟小车返回起点处，观察 Q4.1 的线圈是否断电，小车停止左行。

4．较复杂的自动往返小车控制仿真练习

PLC 的外部接线图与图 3-39 相同，小车用起动按钮起动后，碰到右限位开关 I0.4 停止右行，延时 5s 后自动左行；碰到左限位开关 I0.3 停止左行，延时 7s 后自动右行。按下停止按钮后，小车停止运动。

将程序和系统数据下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。直接使用 PLCSIM 的视图对象或程序状态监控功能调试程序，调试的方法和步骤由读者确定。

注意调试时左、右限位开关接通的时间应大于对应的定时器定时的时间。小车离开某一限位开关后，应将该限位开关对应的输入点置为 0 状态。

3.2.5　实训十一　计数器指令的基本功能

1．计数器的存储器区

每个计数器有一个 16 位的字和一个二进制位，计数器的字用来存放它的当前计数值，计数器触点的状态由它的位的状态来决定。S7-300 的计数器个数（128～2048 个）与 CPU 的型号有关，S7-400 CPU 有 2048 个计数器。

计数器字的 0～11 位是计数值的 BCD 码（见图 3-42），计数值的范围为 0～999。用格式 C#表示计数器的设定值。

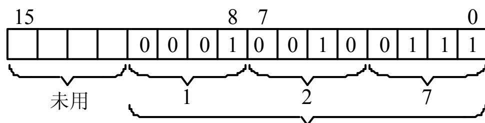


图 3-42　计数器字

在梯形图中计数器有两种表示方法，即方框型的计数器和计数器线圈。

2．加计数器与减计数器

用“新建项目”向导生成一个名为“计数器”的项目（见随书光盘中的同名例程），

CPU 为 CPU 312C。打开 OB1，选择编程语言为梯形图。打开左边的指令列表窗口中的“计数器”文件夹，将其中的 S_CU（加计数器）指令拖放到梯形图中。

图 3-43 和图 3-44 的计数器指令框中，S 为计数器的设置输入端，PV 为预置值输入端，CU 和 CD 分别是加、减计数脉冲输入端，R 为复位输入端；Q 为计数器位输出端，CV 端输出十六进制格式的当前计数值，CV_BCD 端输出当前计数值的 BCD 码。

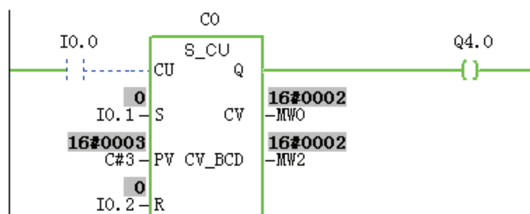


图 3-43 加计数器

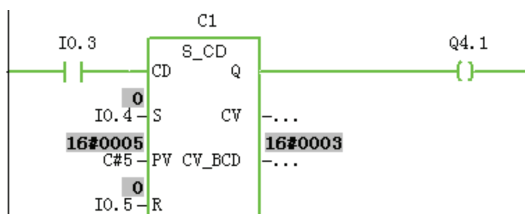


图 3-44 减计数器

CU、CD、S、R、Q 均为 BOOL（位）变量，PV、CV 和 CV_BCD 均为 WORD（字）变量。各变量均可以使用 I、Q、M、L、D 存储区，PV 还可以使用计数器常数 C#。

在“设置”输入信号 S 的上升沿，将 PV 端指定的预置值送入计数器字。在加计数脉冲 CU 的上升沿，如果计数值小于 999，加计数器的计数值加 1。在减计数脉冲 CD 的上升沿，如果计数值大于 0，减计数器的计数值减 1。

复位输入信号 R 为 1 状态时，计数器被复位，计数器位变为 0 状态，计数值被清零。

计数值大于 0 时，计数器的输出 Q 为 1 状态；计数值为 0 时，Q 为 0 状态。

3．加计数器的仿真实验

1）将图 3-43 ～ 图 3-45 中的计数器电路输入到 OB1 中，将 OB1 下载到仿真 PLC，将仿真 PLC 切换到 RUN 或 RUN-P 模式。

2）打开 OB1，单击工具栏上的按钮 ，起动程序状态监控功能。

3）单击 PLCSIM 中加计数脉冲 I0.0 对应的小方框，方框中出现“√”，CV 和 CV_BCD 输出端的计数值加 1 后变为 1，指令框变为绿色，Q4.0 的线圈通电，表示 C0 的计数器位为 1 状态。再单击一次该方框，“√”消失，I0.0 变为 0 状态，计数值不变。多次单击 I0.0 对应的小方框，在 I0.0 由 0 状态变为 1 状态的上升沿，C0 的当前值加 1。

4）用“设置”输入 S 设置计数器时，如果加计数输入信号 CU 为 1 状态，即使 CU 没有变化，下一扫描周期也会加计数。

分别令加计数输入 I0.0 为 0 状态和 1 状态，单击两次 S（设置）输入 I0.1 对应的小方框，观察在 I0.1 的上升沿，CV 和 CV_BCD 输出端的值。

5）在计数器字的值非零时，令复位输入信号 I0.2 为 1 状态，观察复位的效果，计数器值是否变为 0，Q4.0 的线圈是否断电。

4．减计数器的仿真实验

加、减计数器的仿真实验过程基本上相同，做实验时需要注意在减计数信号 CD 的上升沿，计数器值是否减 1。减至 0 时 Q 输出端是否变为 0 状态，Q4.1 的线圈断电。

用“设置”输入 S 设置计数器时，如果减计数输入信号 CD 为 1 状态，即使 CD 没有变化，下一扫描周期也会减计数。



分别令减计数输入 I0.3 为 0 状态和 1 状态，单击两次 S（设置）输入 I0.4 对应的小方框，观察在 I0.4 的上升沿，CV 和 CV_BCD 输出端的值。

计数器一般用来在计了预置值指定的脉冲数后，进行某种操作。为了实现这一要求，最简单的方法是首先将预置值送入减计数器，计数值减为 0 时，其常闭触点闭合，用它来完成要做的工作。如果使用加计数器，则需要增加一条比较指令来判断计数值是否等于预置值。

5．加减计数器的仿真练习

图 3-45 中的加减计数器（S_CUD）的 CU 和 CD 分别是加、减计数输入端。按下面的步骤进行实验：

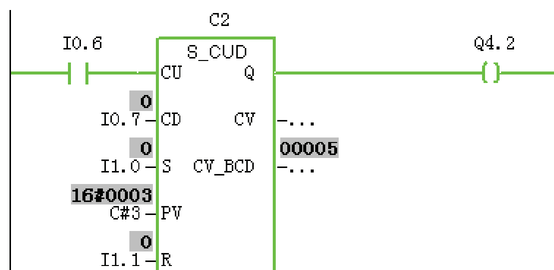


图 3-45 加减计数器

1）观察在 I0.6 和 I0.7 的上升沿，计数值是否分别被加、减 1。

2）观察在 S（设置）输入 I1.0 的上升沿，是否能将 PV 的设定值送给计数器。

3）观察计数值大于 0 和等于 0 时输出信号 Q 的状态。

4）复位输入 I1.1 为 1 时，观察计数值和位输出 Q 的变化。

6．加计数器线圈指令的仿真练习

图 3-46 是用计数器线圈指令设计的加计数器。“设置计数值”线圈 SC 用来设置计数值，图中 I1.2 的常开触点由断开变为接通时，预置值 3 被送入 C3 的计数器字。

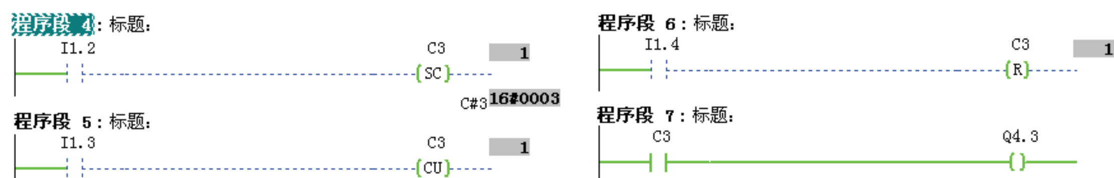


图 3-46 加计数器线圈指令

图中标有 CU 的线圈为加计数器线圈。在 I1.3 的上升沿，如果计数值小于 999，计数值加 1。复位输入 I1.4 为 1 时，计数器被复位，计数器位和计数值被清零。

图 3-46 和图 3-43 中的加计数器的仿真实验过程和仿真的结果相同。

7．计数器线圈指令的仿真练习

1）仿照图 3-46，用减计数器线圈指令（线圈中的符号为 CD）设计一个减计数器。输入到 OB1 后下载到仿真 PLC，验证计数器的功能。

2）对同一个计数器同时使用加计数器线圈指令和减计数器线圈指令，设计一个加减计数器。输入到 OB1 后下载到仿真 PLC，验证计数器的功能。

8．定时器与计数器的仿真练习

按下按钮 I0.0 后，Q0.0 变为 1 状态并自保持(见图 3-47)，I0.1 输入 3 个脉冲后(用 C0 计数)，T0 开始定时，5s 后 Q0.0 变为 0 状态，同时 C0 被复位，用定时器指令和减计数器指令设计出梯形图程序。输入到 OB1 后下载到仿真 PLC，检查是否能满足要求。

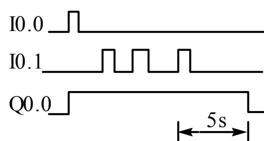


图 3-47　波形图

3.3　数据处理指令与逻辑控制指令

3.3.1　实训十二　数据处理基础知识的仿真实验

1．语句表程序

梯形图是用得最多的 PLC 编程语言。梯形图与继电器电路图相似，直观易懂，很容易被工厂的电气人员掌握，特别适合于数字量逻辑控制。

指令表（西门子称为语句表）是一种文本语言，可以实现某些不能用梯形图实现的功能。在阅读生产中的实际程序时，一般都会遇到语句表程序。

初学者往往觉得语句表语言难学，主要是因为语句表涉及的计算机基础知识较多。只要掌握了有关的知识，多学多练，语句表也不难学。

2．多位二进制数

计算机和 PLC 用多位二进制数来表示数字，二进制数遵循逢 2 进 1 的运算规则，从右往左的第 n 位（最低位为第 0 位）为 1 时，其值为 2^n。二进制数 2#1010 对应的十进制数可以用下式计算：

$$1\times2^3+0\times2^2+1\times2^1+0\times2^0=8+2=10$$

表 3-5 给出了不同进制的数的表示方法，BCD 码的应用将在 3.3.6 节介绍。

表 3-5　不同进制的数的表示方法

十进制数	十六进制数	二进制数	BCD 码	十进制数	十六进制数	二进制数	BCD 码
0	0	00000	0000 0000	9	9	01001	0000 1001
1	1	00001	0000 0001	10	A	01010	0001 0000
2	2	00010	0000 0010	11	B	01011	0001 0001
3	3	00011	0000 0011	12	C	01100	0001 0010
4	4	00100	0000 0100	13	D	01101	0001 0011
5	5	00101	0000 0101	14	E	01110	0001 0100
6	6	00110	0000 0110	15	F	01111	0001 0101
7	7	00111	0000 0111	16	10	10000	0001 0110
8	8	01000	0000 1000	17	11	10001	0001 0111

3．十六进制数

多位二进制数的书写和阅读很不方便。为了解决这一问题，可以用十六进制数取代二进制数。十六进制数的 16 个数字符号是 0～9 和 A～F（对应于十进制数 10～15，见表 3-5）。4 位二进制数对应于 1 位十六进制数，例如二进制数 2#1010 1110 0111 0101 可以转换为 16#AE75。

十六进制数采用逢 16 进 1 的运算规则，从右往左第 n 位的权值为 16^n（最低位的 n 为 0），例如 16#2F 对应的十进制数为 $2\times16^1+15\times16^0=47$。



4．常用的数据类型

在符号表、数据块和块的局部变量表中定义变量时，需要指定变量的数据类型。

表 3-6 中是常用的数据类型，有符号整数的最高位为符号位，正数和负数的符号位分别为 0 和 1。前 8 种数据类型用得较多，TIME_OF_DAY 是 24 小时格式的实时时间。

表 3-6　常用数据类型

数据类型	描述	位数	常数举例
BOOL	二进制位	1	TRUE/FALSE
BYTE	字节	8	B#16#2F
WORD	无符号字	16	W#16#247D
INT	十进制有符号整数	16	−362
DWORD	无符号双字	32	DW#16#149E857A
DINT	十进制有符号双整数	32	L#23
REAL	IEEE 浮点数	32	20.0
S5TIME	S7 时间	16	S5T#1H3M50S
TIME	IEC 时间	32	T#1H3M50S
DATE	IEC 日期	16	D#2009-7-17
TIME_OF_DAY	实时时间	32	TOD#1:10:30.3
CHAR	ASCII 字符	8	‘2A’

一个字节由 8 个位数据组成，例如输入字节 IB3 由 I3.0～I3.7 这 8 位组成（见图 3-48）。

字节、字和双字之间的关系见图 3-49，其中的第 0 位为最低位。

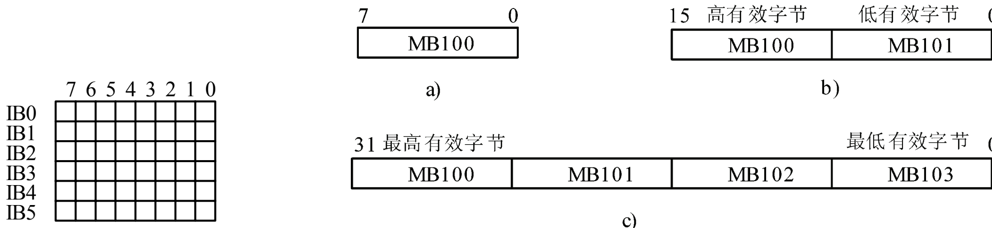


图 3-48　位数据

图 3-49　字节、字和双字

a) MB100　b) MW100　c) MD100

相邻的两个字节组成一个字，相邻的两个字组成 1 个双字。需要注意下面两点：

1）用组成字和双字的编号最小的字节 M100 的编号作为字 MW100 和双字 MD100 的编号。

2）组成字和双字的编号最小的字节 M100 是字 MW100 和双字 MD100 的最高位字节。

STEP 7 用十进制小数来输入或显示浮点数，例如 50 是整数，而 50.0 为浮点数。

5．生成变量表

变量表用来集中监控指定的变量。用鼠标右键单击 SIMATIC 管理器左边窗口中的“块”，执行出现的快捷菜单中的命令“插入新对象”→“变量表”，出现“属性 – 变量表”对话框，生成的变量表默认的名称为“VAT_1”。单击“确定”按钮，VAT_1 被自动打开（见图 3-50）。

在第一行的“地址”列输入 MW8，用右键单击“显示格式”列，执行出现的菜单中的命令，将显示格式修改为 BIN（二进制，见图 3-50）。

在第二行的“地址”列输入 MW8，显示格式被自动设置为 HEX（十六进制）。

在第三行的“地址”列输入 MW8，将显示格式修改为 DEC（十进制）。

用同样的方法输入图 3−50 其他需要监控的变量，双字 MD10 的显示格式为浮点数。

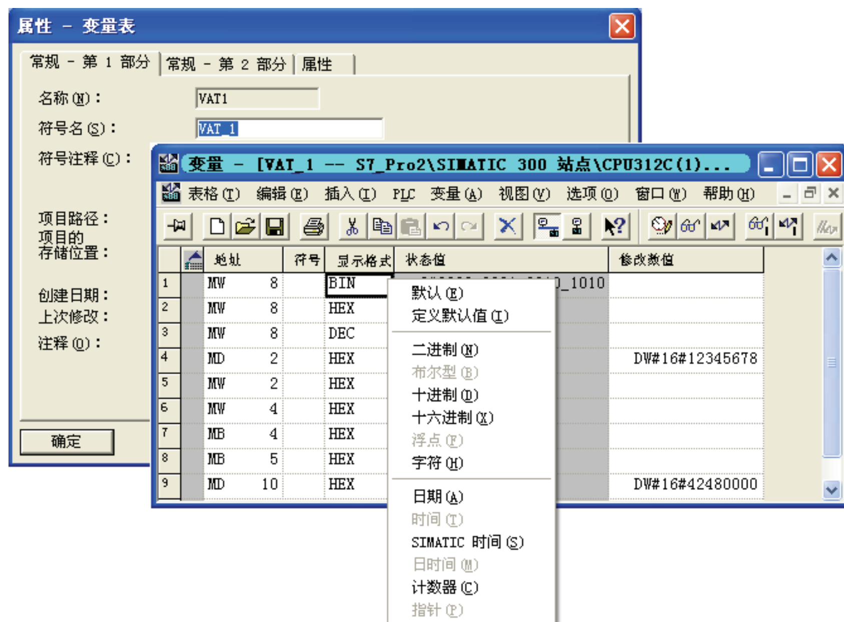


图 3−50　变量表

6. 编写将 MW8 加 1 的程序

打开 HW Config，双击机架中 CPU 模块所在的行，打开 CPU 的“属性”对话框的“周期/时钟存储器”选项卡（见图 3−15）。选中“时钟存储器”多选框，设置时钟存储器（M）的字节地址为 0。M0.7 的周期为 2s（见表 3−3）。

在 OB1 中编写图 3−51 中的程序，每 2s 将 MW8 加 1。ADD_I 是整数加法指令，IN1 与 IN2 相加，结果送回 MW8。

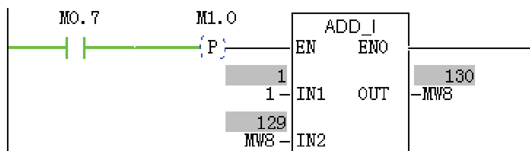


图 3−51　梯形图程序状态

7. 用变量表监控 PLC 中的变量

打开 PLCSIM，选中 SIMATIC 管理器左边窗口中的“块”，将程序块和系统数据下载到仿真 PLC，将仿真 PLC 切换到 RUN−P 模式。双击打开变量表，单击工具栏上的“监视变量”按钮，起动监控功能（见图 3−52）。“状态值”列显示的是 PLC 中的变量值。

MW8 的初始值为 0，每 2s 它被加 1。从变量表的第 1~3 行可以看出二进制数、十六进制数和十进制数之间的关系。

在第 4 行的“修改数值”列输入双字常数 DW#16#12345678，在第 9 行的“修改数值”



列输入浮点数 50.0。单击工具栏上的“激活修改数值”按钮，“修改数值”被写入 PLC，并在“状态值”列显示出来。从第 4～6 行可以看出双字与组成它的两个字之间的关系，从第 6～8 行可以看出字与组成它的两个字节之间的关系。从第 9 行可以看出浮点数用十进制小数输入和显示。

变量 - [VAT_1 -- @S7_Pro2\SIMATIC 300 站点\CPU312C(1)...

表格(T)　编辑(E)　插入(I)　PLC　变量(A)　视图(V)　选项(O)　窗口(W)　帮助(H)

	地址		符号	显示格式	状态值	修改数值
1	MW	8		BIN	2#0000_0100_0000_1110	
2	MW	8		HEX	W#16#040E	
3	MW	8		DEC	1038	
4	MD	2		HEX	DW#16#12345678	DW#16#12345678
5	MW	2		HEX	W#16#1234	
6	MW	4		HEX	W#16#5678	
7	MB	4		HEX	B#16#56	
8	MB	5		HEX	B#16#78	
9	MD	10		FLOATING_POINT	50.0	50.0

图 3-52　变量表

如果仿真 PLC 运行在 RUN 模式，将“修改数值”列的值写入 PLC 时，将会出现“(DOA1)功能在当前保护级别中不被允许”的对话框，必须将仿真 PLC 切换到 RUN-P 模式，才能修改 PLC 中的数据。

8．仿真练习

在变量表中生成用二进制格式显示的 MW20，在 PLCSIM 中生成视图对象 MB20，令其中的 MB20.0 为 1 状态。在变量表中观察 MB20.0 是 MW20 的第几位（最低位为第 0 位），并解释原因。

3.3.2　实训十三　数据传送指令与语句表程序状态监控

1．累加器

32 位累加器用于处理字节、字或双字。几乎所有语句表的操作都是在累加器中进行的，因此需要用装入指令把操作数送入累加器，在累加器中进行运算和数据处理后，用传送指令将累加器 1 中的运算结果传送到某个地址。S7-300 有两个累加器（ACCU1 和 ACCU2），S7-400 有 4 个累加器（ACCU1～ACCU4）。处理 8 位或 16 位数据时，数据存放在累加器的低 8 位或低 16 位。梯形图程序不使用累加器。

2．装入指令与传送指令

装入（L，Load）指令将源操作数装入累加器 1，在此之前，累加器 1 原有的数据被自动移入累加器 2。

装入指令可以对字节（8 位）、字（16 位）和双字（32 位）进行操作，数据长度小于 32 位时，数据在累加器中右对齐，即被装入的数据放在累加器的低端，其余的高位字节填 0。

传送（T，Transfer）指令将累加器 1 的内容写入目的存储区，累加器 1 的内容不变。被复制的数据字节数取决于目的地址的数据长度。

在语句表程序中，存储区之间或存储区与过程映像输入/过程映像输出之间不能直接进行数据交换，累加器相当于上述数据交换的中转站或中间商。

生成一个项目，打开 OB1，执行菜单命令“视图”→“STL”，切换到语句表方式，输入图 3-53 左边的语句表程序。其中的指令“+I”将累加器 1 和累加器 2 中的 16 位整数相加，结果在累加器 1 中。打开 PLCSIM，生成 MW2、MW4 和 MW6 的视图对象。将 OB1 下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。分别将 300 和 500 输入 MW2 和 MW4 的视图对象。

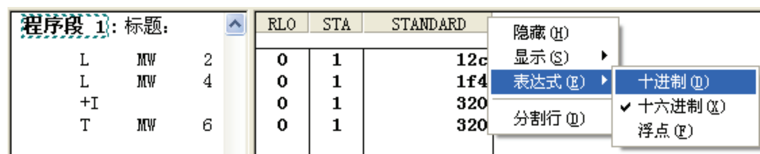


图 3-53　语句表程序状态监控

打开 OB1，单击工具栏上的按钮，起动程序状态监控功能，图 3-53 的右边窗口中是指令执行的监控信息，称为状态域。图中的 RLO 和 STA 是状态字中的两位，将在下一实训中介绍。STANDARD 是累加器 1，默认的显示方式为十六进制数。

用右键单击 STANDARD 所在的表头，执行快捷菜单中的“表达式”→“十进制”命令，改用十进制数显示累加器 1 的值（见图 3-53）。

执行快捷菜单中的“显示”→“累加器 2”命令，监控累加器 2（ACCU 2）的值（见图 3-54）。在快捷菜单中，累加器 1 被称为“默认状态”。

用右键单击某一列，执行快捷菜单中的“隐藏”命令，将使该列消失。

从图 3-54 可以看出，执行第一条 L 指令后，MW2 中的 300 被装入累加器 1，执行第二条指令后，累加器 1 中的 300 被传送到累加器 2，MW4 中的 500 被装入累加器 1。执行“+I”指令后，累加器 1 和累加器 2 中的低位字中的数据相加，运算结果 800 在累加器 1 中。T 指令执行后，累加器 1 中的 800 被传送到 MW6，累加器中的数据不变。

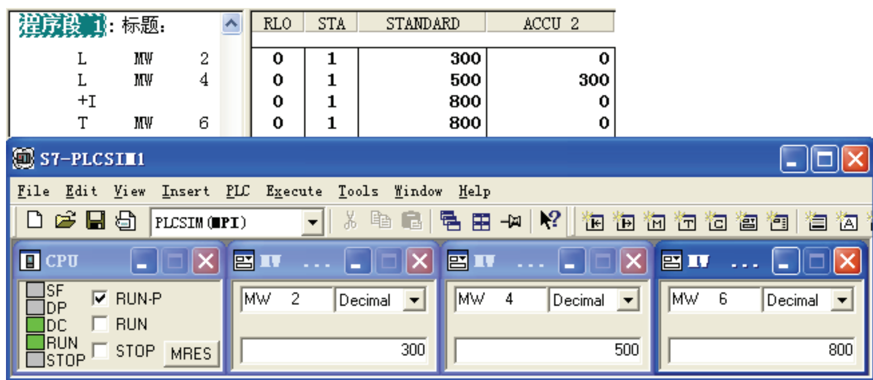


图 3-54　语句表程序状态监控与 PLCSIM

3. 梯形图中的传送指令

梯形图的传送指令（见图 3-55）只有一条 MOVE 指令，它直接将源数据传送到目的地址，不需经过累加器中转。输入变量和输出变量可以是 8 位、16 位或 32 位的基本数据类型。同一条 MOVE 指令的输入变量和输出变量的数据类型可以不相同。实验步骤如下：

1）打开 OB1，执行菜单命令“视图”→“LAD”，切换到梯形图语言，输入图 3-55 中



的梯形图程序。打开 PLCSIM，将 OB1 下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。

2）打开 OB1，单击工具栏上的按钮，起动程序状态监控功能。

3）用右键单击显示的监控数值，执行快捷菜单命令“表达式”→“十进制”，改用十进制数显示监视值。

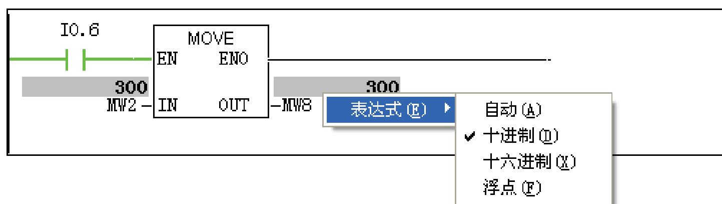


图 3-55　传送指令

4）用 PLCSIM 将 I0.6 置 1，观察指令执行的情况。

5）将 MW8 改为 MB8，下载到 OB1 后，将显示格式改为十六进制，将 MW10 中大于 255 的数传送到 MB8，观察传送的结果。

6）修改程序，将 MB10 中的数传送到 MW6，下载后观察传送的结果。

3.3.3　实训十四　状态字的仿真实验

1．与逻辑运算有关的状态位

状态字是一个 16 位的寄存器，只使用了其中的 9 位（见图 3-56），它用于储存 CPU 执行指令后的状态。

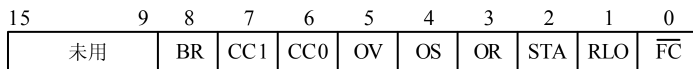


图 3-56　状态字的结构

（1）首次检测位

状态字的第 0 位（最低位）称为首次检测位（$\overline{FC}$）。若该位的状态为 0，表示一个梯形逻辑程序段的开始，或指令为逻辑串的第一条指令。在逻辑串指令执行过程中该位为 1，输出指令或与逻辑运算有关的跳转指令将该位清零。该位只是用于监控。

（2）逻辑运算结果（RLO）

状态字的第 1 位称为逻辑运算结果（Result of Logic Operation，RLO，见图 3-56），它用来存储执行位逻辑指令或比较指令的结果。在梯形图中，RLO 为 1 表示有能流流过指令对应的触点或输出元件，反之则表示没有能流流过。执行完图 3-58 中的第一条指令后，RLO 为 1，梯形图中的 A 点有能流流过（见图 3-57）。执行完第 2 条指令后，RLO 为 0，梯形图中的 B 点没有能流流过。

（3）状态位（STA）

状态字的第 2 位 STA 称为状态位，执行位逻辑指令时，STA 位与指令中的位变量的值一致。由图 3-57 可知，I0.4、I0.6 和 Q4.2 为 1 状态，它们对应的指令执行完后，STA 位为 1 状态。I0.7 和 I0.5 为 0 状态，它们对应的指令执行完后，STA 位为 0 状态。

（4）或位（OR）

状态字的第 3 位称为或位，在先逻辑“与”后逻辑“或”的逻辑运算中，OR 位用来暂存逻辑“与”的操作结果。编程时并不直接使用 OR 位。图 3-57 中 B 点的“与”运算的 RLO 被保存在 OR 位，用来与下面的串联电路的“与”运算的结果作“或”逻辑运算。

2．与逻辑运算有关的状态位的仿真实验

实验步骤如下：

1）生成一个新的项目，将图 3-57 中的梯形图程序输入 OB1。梯形图对应的逻辑代数表达式为 I0.4* I0.7 + I0.6* $\overline{\text{I0.5}}$ = Q4.2，其中的“*”号表示逻辑与，“+”号表示逻辑或，I0.5 上面的水平线表示“非”运算，等号表示将逻辑运算结果赋值给 Q4.2。

2）执行菜单命令“视图”→“STL”，切换到语句表方式。图 3-58 的左边是图 3-57 中的梯形图对应的语句表指令。语句中的 A 和 AN 分别表示串联的常开触点和常闭触点，O 表示两条串联电路的并联，等号表示赋值。

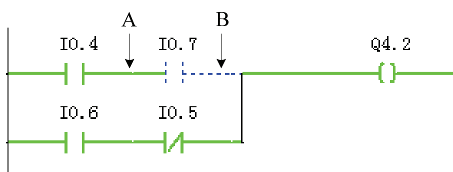


图 3-57 梯形图

程序段 3：标题:

语句			RLO	STA	STATUS WORD
A	I	0.4	1	1	1_0000_0111
A	I	0.7	0	0	1_0000_0001
O			0	1	1_0000_0100
A	I	0.6	1	1	1_0000_0111
AN	I	0.5	1	0	1_0000_0011
=	Q	4.2	1	1	1_0000_0110

图 3-58 语句表程序状态监控

3）打开 PLCSIM，将 OB1 下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。

4）单击工具栏上的按钮，起动程序状态监控功能。用实训十三所述的方法，隐藏累加器 1（STANDARD），增加对状态字（STATUS WORD）的监控。

5）用 PLCSIM 的视图对象 IB0 改变 I0.4 的状态，观察第一条指令的 STA 位是否随之而变。用同样的方法观察其他 A 指令和 AN 指令的 STA 位与指令中的输入点状态的关系。

6）改变 I0.4～I0.7 的状态，观察对 RLO 的影响，RLO 是否反映了程序中的逻辑运算结果？

7）改变 I0.4 或 I0.7 的状态，观察 B 点的 RLO 与 OR 位的关系。

8）改变 I0.4~I0.7 的状态，观察最后一条指令执行完后（即开始执行下一个梯形逻辑程序段时），状态字的最低位 $\overline{\text{FC}}$ 位是否为 0。

3．溢出位与溢出保持位的仿真实验

状态字的第 5 位和第 4 位分别是溢出位 OV 和溢出保持位 OS。如果执行数学运算指令时出现错误，例如溢出（运算结果超出允许的范围）、非法操作和不规范的格式，溢出位被置 1。如果后面影响该位的指令的执行没有出错，该位被清零。

溢出状态保持位 OS 用来保存 OV 位，只有 JOS（OS＝1 时跳转）指令、块调用指令和块结束指令才能复位 OS 位。实际上很少使用 OS 位。

将图 3-59 中的语句表程序输入到 OB1，指令“*I”是 16 位整数乘法指令，如果乘积超出允许的范围，执行该指令后，OV 位和 OS 位（状态字的第 5 位和第 4 位）均为 1。

将 OB1 下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。打开 OB1，单击工具栏上的按钮，起动程序状态监控功能。用 PLCSIM 修改 MW10 的值。图 3-59 和图 3-60 分别



是没有溢出和有溢出的监控结果。请注意状态字的第 4、第 5 位。有溢出（运算结果大于 32767）时执行 JO 指令，将跳转到标号 OVER 处。跳转指令将在实训十五中介绍。

程序段 2: 标题:

语句	STANDARD	ACCU 2	STATUS WORD
L 3200	3200	0	0_0000_0000
L MW 10	10	3200	0_0000_0000
*I	32000	0	0_1000_0000
JO OVER	32000	0	0_1000_0000
L 250	250	32000	0_1000_0000
T MW 12	250	32000	0_1000_0000
OVER: A I 2.1	250	32000	0_1000_0001

图 3-59　没有溢出的语句表程序状态监控

程序段 2: 标题:

语句	STANDARD	ACCU 2	STATUS WORD
L 3200	3200	0	0_0000_0000
L MW 10	11	3200	0_0000_0000
*I	35200	0	0_1011_0000
JO OVER	35200	0	0_1011_0000
L 250	250	32000	0_1000_0000
T MW 12	250	32000	0_1000_0000
OVER: A I 2.1	35200	0	0_1011_0001

图 3-60　有溢出的语句表程序状态监控

4. 用在线帮助获取指令对状态字的影响

选中图 3-60 中的“A”指令，按计算机的〈F1〉键，打开该指令的在线帮助，可以看到该指令的执行对状态字各位的影响。由图 3-61 可知，该指令使状态字的 $\overline{\text{FC}}$ 位为 1，OR、STA 和 RLO 位受该指令执行的影响，其他位不受影响。

BR	CC 1	CC 0	OV	OS	OR	STA	RLO	/FC
-	-	-	-	-	x	x	x	1

图 3-61　在线帮助中指令执行对状态字的影响

5. 条件码

状态字的第 7 位和第 6 位称为条件码 1（CC1）和条件码 0（CC0）。这两位综合起来，用于表示在累加器 1 中执行的数学运算或字逻辑运算的结果与 0 的大小关系、比较指令的执行结果或移位指令的移出位状态。用户程序一般并不直接使用条件码。

指令的在线帮助给出了条件码与上述指令的执行结果的关系。执行完图 3-60 中的乘法指令“*I”后，CC1 和 CC0 分别为 1 和 0，表示运算结果为正。

6. 二进制结果位的仿真实验

状态字的第 8 位为二进制结果位 BR。在梯形图中，用方框表示某些指令、功能（FC）和功能块（FB）。图 3-62 中 I1.0 的常开触点接通时，能流流到方框指令 DIV_I 的数字量输入端 EN（Enable，使能输入），该指令才能执行。

如果方框指令的 EN 端有能流流入，而且执行时无错误，则使能输出 ENO（Enable Output）端有能流传递给下一个元件（见图 3-62）。如果指令执行出错（除数为零），能流在出现错误的方框指令终止（见图 3-63）。ENO 可以作为下一个方框的 EN 输入，即几个方框可以串联，只有前一个方框被正确执行，与它连接的后面的程序才能被执行。

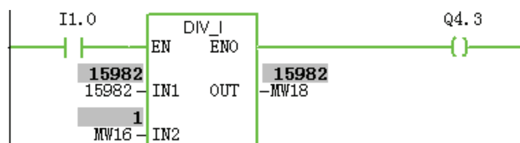


图 3-62　BR 位为 1

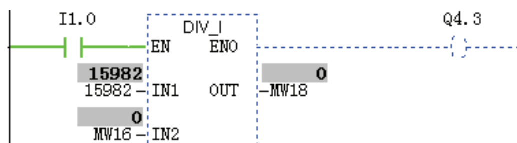


图 3-63　BR 位为 0

状态字中的二进制结果位 BR 对应于方框指令的 ENO，如果指令被正确执行，BR 位为 1，ENO 端有能流流出。如果执行出现错误，BR 位为 0，ENO 端没有能流流出。

仿真实验步骤如下：

1）将图 3-62 中的程序输入到 OB1。

2）将 OB1 下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。

3）打开 OB1，单击工具栏上的按钮，起动程序状态监控功能。

4）用 PLCSIM 设置 MW16 中的除数非零，将 I1.0 置为 1 状态，能流流入方框指令 DIV_I 的 EN 输入端。因为指令执行成功，指令框为绿色，有能流从它的使能输出端 ENO 流出（见图 3-62）。

5）设置除数 VW16 的值为 0，指令执行出错，DIV_I 指令框变为虚线，没有能流从它的 ENO 输出端流出（见图 3-63）。

3.3.4 实训十五　逻辑控制指令的仿真实验

1．逻辑控制指令

逻辑控制指令的执行取决于当时有关的状态位的状态。指令表中的逻辑控制指令（见表 3-7）包括跳转指令和循环指令。在没有执行跳转指令和循环指令时，各条语句按从上到下的先后顺序逐条执行。执行逻辑控制指令时（不包括无条件跳转指令），根据状态字中有关位的状态，决定是否跳转到指令中的地址标号所在的目的地址。跳转到目的地址后，程序继续顺序执行。

表 3-7　逻辑控制指令与状态位触点指令

逻辑控制指令	状态位触点指令	描　述	逻辑控制指令	状态位触点指令	描　述
JU	—	无条件跳转到标号指定的目的地址	JOS	OS	OS=1 时跳转或触点闭合
JL	—	多分支跳转	JZ	== 0	运算结果为 0 时跳转或触点闭合
JC	—	RLO=1 时跳转到标号指定的目的地址	JN	<> 0	运算结果非 0 时跳转或触点闭合
JCN	—	RLO=0 时跳转到标号指定的目的地址	JP	> 0	运算结果为正时跳转或触点闭合
JCB	—	RLO=1 时跳转，将 RLO 复制到 BR	JM	< 0	运算结果为负时跳转或触点闭合
JNB	—	RLO=0 时跳转，将 RLO 复制到 BR	JPZ	>= 0	运算结果>=0 时跳转或触点闭合
JBI	BR	BR=1 时跳转或梯形图中触点闭合	JMZ	<= 0	运算结果<=0 时跳转或触点闭合
JNBI	—	BR=0 时跳转	JUO	UO	指令出错时跳转或触点闭合
JO	OV	OV=1 时跳转或触点闭合	LOOP	—	循环指令

只能在同一个逻辑块内跳转，在一个块内，同一个跳转目的地址只能出现一次。

跳转或循环指令的操作数为地址标号，标号由最多 4 个字符组成，第一个字符必须是字母，其余的可以是字母、数字和下划线。在语句表中，目标标号与目标指令用冒号分隔。

2．语句表中的逻辑控制指令的应用

要求在整数除法运算的除数为 0 时，将错误标志 M0.0 置位，除数非 0 时将 M0.0 复位。

下面是根据要求编写的语句表程序，图 3-64 是程序的流程图。

```
L      1600           //常数装入累加器 1 的低字
L      MW     16      //累加器 1 的内容装入累加器 2，MW16 的内容装入累加器 1
/I                    //累加器 2 的值除以累加器 1 的值
T      MW     18      //将累加器 1 中的商传送到 MW18
JUO    M001           //如果累加器 1 的除数为 0（运算出错），则跳转到标号 M001 处
CLR                   //除数非零时复位 RLO
```



```
        =     M       0.0            //复位错误标志位
        JU    M002                   //无条件跳转到标号 M002 处
M001: SET                            //除数为零时置位 RLO
        =     M       0.0            //置位错误标志位
M002: NOP     0                      //跳转到此标号后，继续执行程序扫描
```

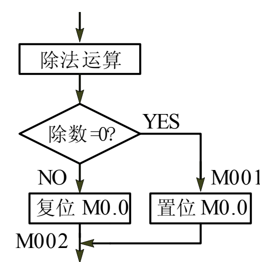


图 3-64　跳转指令流程图

程序段 1：标题：	RLO	STANDARD	STATUS WORD
L 1600	0	1600	0_0000_0100
L MW 16	0	0	0_0000_0100
/I	0	0	0_1111_0100
T MW 18	0	0	0_1111_0100
JUO M001	0	0	0_1111_0100
CLR	0	800	0_1000_0000
= M 0.0	0	800	0_1000_0000
JU M002	0	800	0_1000_0000
M001: SET	1	0	0_1111_0110
= M 0.0	1	0	0_1111_0110
M002: NOP 0	1	0	0_1111_0110

图 3-65　除数为 0 的程序状态

打开 PLCSIM，将 OB1 下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。

打开 OB1，单击工具栏上的按钮，起动程序状态监控功能。设置只监控 RLO、累加器 1（STANDARD）和状态字（STATUS WORD）。用鼠标右键单击累加器 1 所在的区域，执行快捷菜单中的“表达式”→“十进制”命令（见图 3-53），设置显示方式为十进制显示。

用 PLCSIM 设置除数 MW16 为 0，执行除法指令出错，跳转到标号 M001 处，将标志 M0.0 置位为 1。被跳过的指令的监控值用普通字体显示（见图 3-65），被执行的指令的监控值用加粗的字体显示。

将除数 MW16 设置为非 0，除法指令被正确执行，不会跳转到标号 M001。M0.0 被复位后，执行无条件跳转指令 JU，跳转到标号 M002 处。程序状态见图 3-66。

程序段 1：标题：	RLO	STANDARD	STATUS WORD
L 1600	0	1600	0_0000_0100
L MW 16	0	2	0_0000_0100
/I	0	800	0_1000_0100
T MW 18	0	800	0_1000_0100
JUO M001	0	800	0_1000_0100
CLR	0	800	0_1000_0000
= M 0.0	0	800	0_1000_0000
JU M002	0	800	0_1000_0000
M001: SET	1	0	0_1111_0110
= M 0.0	1	0	0_1111_0110
M002: NOP 0	0	800	0_1000_0000

图 3-66　除数非 0 的程序状态

3．梯形图中的跳转指令

梯形图中的无条件跳转指令与条件跳转指令的助记符均为 JMP（Jump），无条件跳转指令线圈直接与右边的垂直电源线相连(见图 3-68 的程序段 4)，执行无条件跳转指令后马上跳转到指令给出的标号处。

条件跳转指令的线圈受触点电路的控制，跳转线圈“通电”时，跳转到指令给出的标号处(见图 3-68 的程序段 2)。

用下面的方法生成标号：打开梯形图编辑器左边的指令浏览器窗口的“跳转”文件夹，将其中的“LABEL”（标号）“拖放”到梯形图中。双击其中的“???”，输入标号。标号必须

放在一个程序段开始的地方。

4．梯形图中的状态位触点指令

梯形图中的状态位指令以常开触点或常闭触点的形式出现。这些触点的通断取决于状态位 BR、OV、OS 的状态（见表 3-7），以及数学运算的结果与 0 的关系。CC0 和 CC1 均为 1 时，表示数学运算指令有错误，标有 UO 的常开触点闭合。状态位触点可以与别的触点串并联。

5．梯形图中的状态位触点指令与跳转指令应用举例

图 3-67 是程序执行的流程图，图 3-68 是梯形图程序。整数乘法指令“MUL_I”的运算如果有溢出（乘积大于 16 位整数能表示的最大正数 32767），OV 位的常开触点闭合(见程序段 2)，将跳转到标号 M003 处。如果乘法运算没有溢出，OV 位的常开触点断开，跳转指令线圈断电，不会跳转，将顺序执行程序段 3。执行完后，在程序段 4 无条件跳转到标号 M004 处。

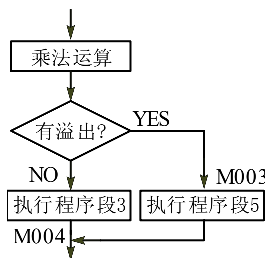


图 3-67　跳转指令流程图

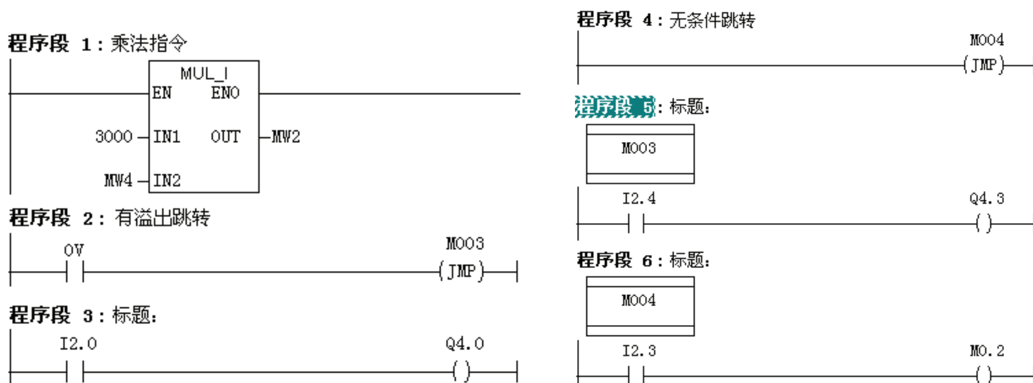


图 3-68　状态位触点指令与跳转指令的应用

6．仿真实验

打开 PLCSIM，生成监控 MW4 的视图对象，将 OB1 下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。

打开 OB1，单击工具栏上的按钮，起动程序状态监控功能。

用视图对象设置 MW4 的值为 8，通过 OV 触点的状态观察乘法运算是否有溢出，是否从程序段 2 跳转到了标号 M003 处？程序段 5 是否被跳过？被跳过的程序段左边的“电源线”没有能流。

用视图对象设置 MW4 的值为 5000，通过 OV 触点的状态观察乘法运算是否有溢出，是否跳过了程序段 3 和程序段 4。

3.3.5　实训十六　比较指令的仿真实验

1．比较指令

比较指令用来比较两个具有相同数据类型的数，可以比较整数（I）、双整数（D）和浮点数（R）。方框比较指令在梯形图中相当于一个常开触点，可以与其他触点串联和并联。在使能输入信号为 1 时，比较 IN1 和 IN2 输入的两个操作数。如果被比较的两个数满足指令指



定的大于、等于、小于等条件，比较结果为“真”，等效触点闭合，指令框有能流流过。

梯形图中比较指令框的输入和输出均为 BOOL 变量，可以取 I、Q、M、L 和 D；被比较数 IN1 和 IN2 的数据长度与指令有关，可以是整数（I）、双整数（D）和浮点数（R）。数据类型为 I、Q、M、L、D 或常数。

表 3-8 中的“?”可以取==（等于）、<>（不等于）、>、<、>=和<=。

表 3-8　比较指令

语句表指令	梯形图中的符号	描　　述
? I	CMP ? I	比较累加器 2 和累加器 1 低字中的整数是否==, <>,>,<,>=,<=，如果条件满足，RLO=1
? D	CMP ? D	比较累加器 2 和累加器 1 中的双整数是否==, <>,>,<,>=,<=，如果条件满足，RLO=1
? R	CMP ? R	比较累加器 2 和累加器 1 中的浮点数是否==, <>,>,<,>=,<=，如果条件满足，RLO=1

2．基于比较指令的方波发生器

图 3-69 中的 T0 是接通延时定时器，I0.0 的常开触点接通时，T0 开始定时，其剩余时间值从预置时间值 2s 开始递减。减至 0 时，T0 的定时器位变为 1 状态，它的常闭触点断开，使它的定时器位变为 0。T0 的常闭触点闭合，又从预置时间值开始定时。

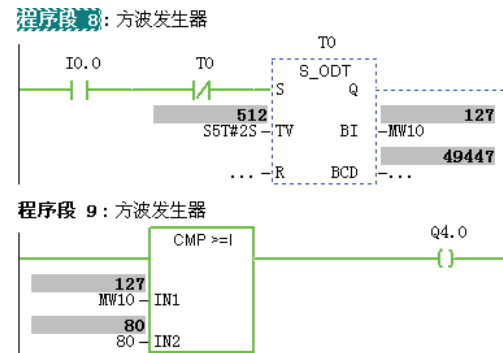


图 3-69　方波发生器电路

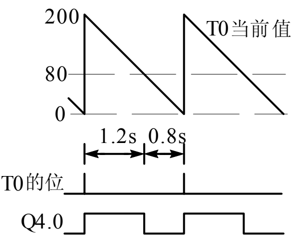


图 3-70　方波发生器的波形图

T0 的十六进制剩余时间（单位为 10ms）被写入 MW10 后，与常数 80 比较。剩余时间大于等于 80（800ms）时，比较指令等效的触点闭合，Q4.0 的线圈通电，通电的时间为 1.2s。剩余时间小于 80 时，比较指令等效的触点断开，Q4.0 的线圈断电 0.8s。

将程序输入到 OB1 后下载到仿真 PLC，起动程序状态监控，接通 I0.0 的常开触点，观察 Q4.0 的状态和 T0 的剩余时间值是否按图 3-70 的波形变化。

语句表中的比较指令用于比较累加器 1 与累加器 2 中数据的大小。如果比较的条件满足，则 RLO 为 1，否则为 0。状态字的 CC0 和 CC1 位用来表示两个数的大于、小于和等于关系。

下面是图 3-69 中的程序段 9 对应的语句表程序：

```
L    MW    10
L    80
>=I
=    Q     4.0
```

3. 路灯控制电路的仿真练习

OB1 的局部变量 OB1_DATE_TIME 是调用 OB1 的日期和时间，共 8 个字节。其数据格式为 DATE_AND_TIME，起始地址为 LB12，8 个字节分别是 BCD 码格式的年的低两位、月、日、时、分、秒、毫秒的百位和十位，最后一个字节的 0～3 位代表星期，4～7 位是毫秒的个位。时、分的值在 LW15 中。

路灯控制电路见图 3-71，LW15 中的时、分值大于等于 16#2000（20: 00）或小于 16#600（6: 00）时，控制路灯的 Q4.1 的线通电，反之则断电。

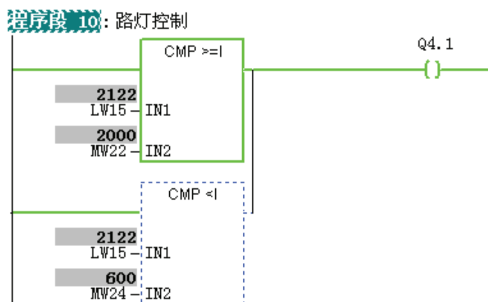


图 3-71　路灯控制程序

将程序输入到 OB1 后下载到仿真 PLC，起动程序状态监控，显示格式为十六进制数（实际上是 BCD 码）。用 PLCSIM 设置 MW22 中的开灯时间和 MW24 中的关灯时间的时、分值。为了节约调试的时间，可将开灯时间和关灯时间设置得距离当前时间尽量近一些。观察是否能按设置的时间开灯和关灯。

3.3.6　实训十七　数据转换指令的仿真实验

数据转换指令将累加器 1 中的数据进行数据类型的转换，转换的结果仍然在累加器 1。数据转换指令见表 3-9。

1. BCD 码

BCD 码是二进制编码的十进制数的缩写，用 4 位二进制数表示一位十进制数（见表 3-5），每位的数值范围为 2#0000～2#1001，对应于十进制数 0～9。

BCD 码的最高 4 位二进制数用来表示符号，负数的最高位为 1，正数为 0，其余 3 位可取 0 或 1。16 位 BCD 码字的范围为–999～+999，32 位 BCD 码双字的范围为–9999999～+9999999。BCD 码各位之间的关系是逢十进一，图 3-72 中的 BCD 码为-862，图 3-73 是 7 位 BCD 码的格式。



图 3-72　3 位 BCD 码的格式

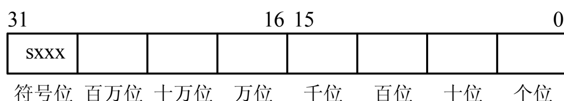


图 3-73　7 位 BCD 码的格式

拨码开关（见图 3-74）的圆盘圆周面上有 0～9 这 10 个数字，用按钮来增、减各位要输入的数字。它用内部的硬件将 10 个数字转换为 4 位二进制数。PLC 用输入点读取的多位拨码开关的输出值就是 BCD 码，需要用数据转换指令将它转换为整数（或双整数）。



用 PLC 的 4 个输出点给译码驱动芯片 4547 提供输入信号（见图 3-75），可以用 LED 七段显示器显示一位十进制数。需要用数据转换指令，将 PLC 中的整数或双整数转换为 BCD 码，然后分别送给各个译码驱动芯片。



图 3-74　拨码开关

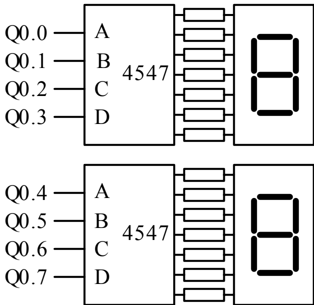


图 3-75　LED 七段显示器电路

表 3-9　数据转换指令

语　句　表	梯　形　图	描　　述
BTI	BCD_I	将累加器 1 低字的 3 位 BCD 码转换成整数
ITB	I_BCD	将累加器 1 低字的整数转换成 3 位 BCD 码
BTD	BCD_DI	将累加器 1 的 7 位 BCD 码转换成双整数
DTB	DI_BCD	将累加器 1 的双整数转换成 7 位 BCD 码
DTR	DI_R	将累加器 1 的双整数转换成浮点数
ITD	I_DI	将累加器 1 低字的整数转换成双整数
RND	ROUND	将浮点数转换为四舍五入的双整数
RND＋	CEIL	将浮点数转换为大于等于它的最小双整数
RND－	FLOOR	将浮点数转换为小于等于它的最大双整数
TRUNC	TRUNC	将浮点数转换为截位取整的双整数
CAW	—	交换累加器 1 低字中两个字节的位置
CAD	—	交换累加器 1 中 4 个字节的顺序

浮点数转换为双整数有 4 条指令，它们将浮点数转换为 32 位双整数。使用得最多的是四舍五入的 ROUND 指令，CEIL 和 FLOOR 指令用得很少。

2．BCD 码与整数相互转换的仿真实验

打开 OB1，执行菜单命令“视图”→“LAD”，切换到梯形图语言，输入图 3-76 和图 3-77 中的梯形图程序。

用鼠标右键单击 SIMATIC 管理器左边窗口中的“块”，执行出现的快捷菜单中的命令“插入新对象”→“变量表”，生成变量表 VAT_1（见图 3-78）。显示格式 HEX 和 DEC 分别是十六进制数和十进制数。

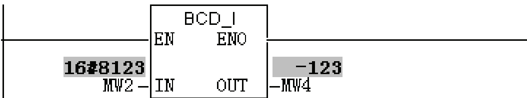


图 3-76　BCD 码转换为整数

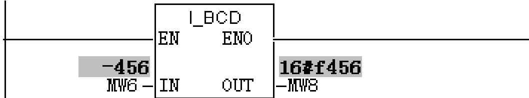


图 3-77　整数转换为 BCD 码

打开 PLCSIM，将 OB1 下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。打开 OB1，单击工具栏上的按钮，起动程序状态监控功能。

在变量表第 1 行的“修改数值”列输入十六进制数 W#16#8123（从拨码开关读取的 BCD 码），二进制数的最高位（符号位）为 1，表示负数。单击工具栏上的按钮，“修改数值”被写入 PLC 内的 MW2，并在“状态值”列显示出来。图 3-76 中的指令将它转换为十进制数-123。

将 MW2 中的数分别修改为 W#16#F123（最高 4 位二进制数均为 1）和 W#16#0123，观察转换后的 MW4 中的十进制数。

VAT_1 -- @S7_Pro1\SIMATI...

	地址		显示格式	状态值	修改数值
1	MW	2	HEX	W#16#8123	W#16#8123
2	MW	4	DEC	-123	
3	MW	6	DEC	-456	-456
4	MW	8	HEX	W#16#F456	

图 3-78　变量表

在变量表第 3 行的“修改数值”列输入十进制数-456，单击工具栏上的“激活修改数值”按钮，“修改数值”被写入 PLC 内的 MW6。图 3-77 中的指令将它转换为 BCD 码 W#16#f456。二进制数的最高 4 位均为 1，表示该数是负数。

将 MW6 中的数修改为 456，观察转换后的 MW8 中的 BCD 码。

执行菜单命令“视图”→“STL”，切换到语句表语言，观察图 3-76 和图 3-77 对应的语句表程序。

3．BCD 码转换的仿真练习

将图 3-76 和图 3-77 中的程序修改为 BCD 码与双字的转换指令，修改变量表中的地址。

打开 PLCSIM，将修改后的程序下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。用变量表设置指令的输入值，观察转换后的结果是否正确。

4．压力计算程序中的数据转换

压力变送器的量程为 0～10MPa，输出信号为 4～20mA，模拟量输入模块的量程为 4～20mA，转换后的数字量为 0～27 648，设转换后的数字为 N，以 kPa 为单位的压力值的转换公式为

$$P = (10000 \times N) / 27648 = 0.36169 \times N \quad \text{(kPa)} \qquad (3\text{-}1)$$

来自 AI 模块的 PIW320 的原始数据为 16 位整数，首先用 I_DI 指令将整数转换为双整数，然后用 DI_R 指令转换为实数（Real），再用实数乘法指令 MUL_R 完成式（3-1）的运算（见图 3-79）。最后用四舍五入的 ROUND 指令，将运算结果转换为以 kPa 为单位的整数。

打开 PLCSIM，将程序下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。将 0 和 27648 分别输入 PIW320，观察 MD24 中的计算结果是否是 0 和 10000 kPa。将 0~27648 之间的任意数值输入 PIW320，观察计算结果是否与计算器计算的相同。

指令 ROUND 的运算结果为双字，但是由式（3-1）可知最终的运算结果实际上不会超过一个字，保存运算结果的 MD20 的高位字 MW20 的值为 0，运算结果的有效部分在低位字 MW22 中。观察 MW20 和 MW22 的值，验证上述结论是否正确。

5．数据转换指令的仿真练习

半径（小于 1000 的整数）在 MW10 中，圆周率为 3.14159，用浮点数运算指令计算圆的周长，将运算结果转换为整数，存放在 MW14 中。将程序输入到 OB1 后下载到仿真 PLC，调试程序直到满足要求。



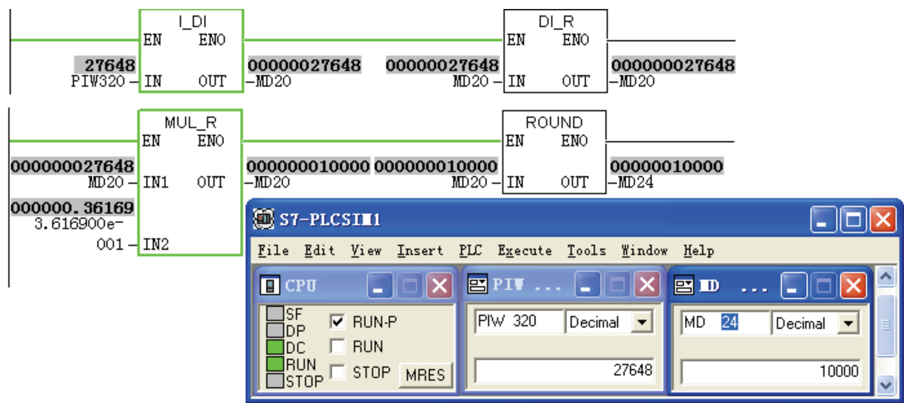


图 3-79　使用浮点数运算指令的压力测量值计算程序

3.4　数学运算指令

3.4.1　实训十八　数学运算指令的仿真实验

数学运算指令分为整数运算指令和浮点数运算指令。

1．整数数学运算指令

整数数学运算指令见表 3-10。

表 3-10　整数数学运算指令

语　句　表	梯　形　图	描　　述
+I	ADD_I	将累加器 1、2 低字的整数相加，运算结果在累加器 1 的低字
–I	SUB_I	累加器 2 低字的整数减去累加器 1 低字的整数，运算结果在累加器 1 的低字
*I	MUL_I	将累加器 1、2 低字的整数相乘，32 位双整数运算结果在累加器 1
/ I	DIV_I	累加器 2 低字的整数除以累加器 1 低字的整数，商在累加器 1 的低字，余数在累加器 1 的高字
+		累加器 1 的内容与 16 位或 32 位常数相加，运算结果在累加器 1
+D	ADD_DI	将累加器 1、2 的双整数相加，32 位双整数运算结果在累加器 1
–D	SUB_DI	累加器 2 的双整数减去累加器 1 的双整数，双整数运算结果在累加器 1
*D	MUL_DI	将累加器 1、2 的双整数相乘，双整数运算结果在累加器 1
/ D	DIV_DI	累加器 2 的双整数除以累加器 1 的双整数，32 位商在累加器 1，余数被丢掉
MOD	MOD_DI	累加器 2 的双整数除以累加器 1 的双整数，32 位余数在累加器 1

2．整数数学运算的仿真实验

改用整数数学运算指令实现（3-1）式的压力运算：

$$P = (10000 \times N) / 27648 \quad (\text{kPa}) \qquad (3\text{-}2)$$

在运算时一定要先乘后除，否则会损失原始数据的精度。整数四则运算指令有 16 位和 32 位两种，应根据指令的输入、输出数据可能的最大值选用适当的指令。

假设用于测量压力的 AI 模块的通道地址为 PIW320。模拟量满量程时 A/D 转换后的数字 N 的值为 27648，乘以 10000 以后乘积可能超过 16 位整数的允许范围，因此应采用双整数的乘法指令 MUL_DI。式（3-2）中的被除数是双整数，因此应采用双整数除法指令 DIV_DI。

首先应使用指令 I_DI 将 PIW320 中的原始数据（16 位整数）转换为双整数（见图

3-80），双字乘、除法指令中的常数应使用“L#”开始的 32 位的双整数常数。

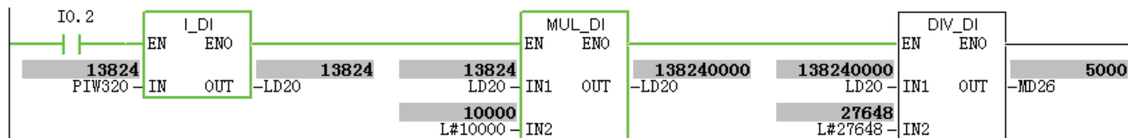


图 3-80 压力测量值计算程序

图 3-80 与图 3-79 中的程序的调试方法相同。

3. 浮点数数学运算指令

浮点数（实数）的数据类型为 REAL，浮点数数学运算指令见表 3-11。浮点数四则运算指令对累加器 1 和累加器 2 中的 32 位 IEEE 格式的浮点数进行运算，运算结果在累加器 1。

表 3-11 浮点数数学运算指令

语 句 表	梯 形 图	描 述
+R	ADD_R	累加器 1、2 的浮点数相加，浮点数运算结果在累加器 1
–R	SUB_R	累加器 2 的浮点数减去累加器 1 的浮点数，浮点数运算结果在累加器 1
*R	MUL_R	累加器 1、2 的浮点数相乘，浮点数乘积在累加器 1
/R	DIV_R	累加器 2 的浮点数除以累加器 1 的浮点数，浮点数商在累加器 1 中，余数被丢掉
ABS	ABS	累加器 1 的浮点数取绝对值，浮点数运算结果在累加器 1
SQR	SQR	求累加器 1 的浮点数的平方，浮点数运算结果在累加器 1
SQRT	SQRT	求累加器 1 的浮点数的平方根，浮点数运算结果在累加器 1
EXP	EXP	求累加器 1 的浮点数的自然指数，浮点数运算结果在累加器 1
LN	LN	求累加器 1 的浮点数的自然对数，浮点数运算结果在累加器 1
SIN	SIN	求累加器 1 的浮点数的正弦函数，浮点数运算结果在累加器 1
COS	COS	求累加器 1 的浮点数的余弦函数，浮点数运算结果在累加器 1
TAN	TAN	求累加器 1 的浮点数的正切函数，浮点数运算结果在累加器 1
ASIN	ASIN	求累加器 1 的浮点数的反正弦函数，浮点数运算结果在累加器 1
ACOS	ACOS	求累加器 1 的浮点数的反余弦函数，浮点数运算结果在累加器 1
ATAN	ATAN	求累加器 1 的浮点数的反正切函数，浮点数运算结果在累加器 1

4. 浮点数数学运算指令应用的仿真练习

浮点数三角函数指令的角度是以弧度为单位的浮点数。图 3-81 的 MD30 中的角度值是以度为单位的浮点数，使用三角函数指令之前应先将角度值乘以π/180.0（0.01745329），转换为弧度值，然后用 SIN 指令求角度的正弦值。

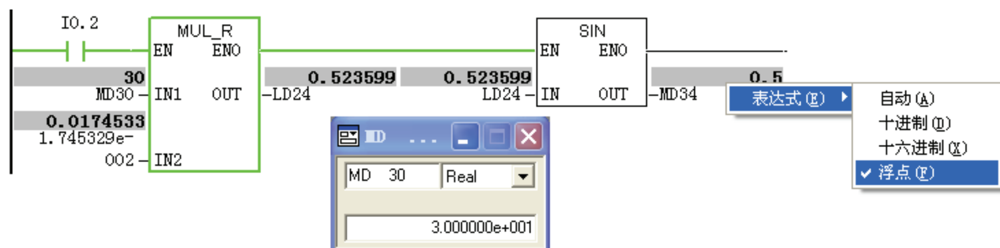


图 3-81 浮点数运算程序



打开 PLCSIM，将程序下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。将 30.0 输入 MD30，观察 MD34 中的计算结果是否是 0.5。输入 0.0~360.0 之间的任意实数，观察 MD34 中的运算结果是否与计算器计算的相同。

5．数学运算指令的仿真练习

某温度变送器的量程为 −100℃～500℃，输出信号为 4～20mA，某模拟量输入模块将 0～20mA 的电流信号转换为数字 0～27648，设转换后得到的数字为 N，求以 0.1℃为单位的温度值。

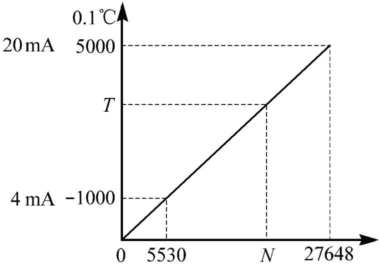


图 3-82　模拟量与转换值的关系

单位为 0.1℃的温度值−1000～5000 对应于数字量 5530～27648，由图 3-82 给出的比例关系可列出下式：

$$\frac{T-(-1000)}{N-5530}=\frac{5000-(-1000)}{27648-5530}$$

经数学运算求出 T 的计算公式为

$$T=\frac{6000\times(N-5530)}{22118}-1000 \qquad (0.1℃)$$

根据上式编写出温度计算程序，打开 PLCSIM，将程序下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。用 PLCSIM 设置输入 N 的值分别为 5530 和 27648，观察输出 T 是否为−1000 和 5000。设置 N 为某个中间值，观察输出 T 是否与计算器算出的相同。

3.4.2　实训十九　移位与循环移位指令的仿真实验

1．移位指令概述

移位指令将累加器 1 的低字或累加器 1 的全部内容左移或右移若干位（见表 3-12）。

无符号数（字或双字）移位后空出来的位填以 0，有符号数（整数或双整数）右移后空出来的位填以符号位对应的二进制数，正数的符号位为 0，负数的符号位为 1。最后移出的位被装入状态字的 CC1 位。

表 3-12　移位指令

语　句　表	梯　形　图	描　　述
SSI	SHR_I	将累加器 1 低字的有符号整数逐位右移，空出的位添上与符号位相同的数
SSD	SHR_DI	将累加器 1 的有符号双整数逐位右移，空出的位添上与符号位相同的数
SLW	SHL_W	将累加器 1 低字的 16 位字逐位左移，空出的位添 0
SRW	SHR_W	将累加器 1 低字的 16 位字逐位右移，空出的位添 0
SLD	SHL_DW	将累加器 1 的双字逐位左移，空出的位添 0
SRD	SHR_DW	将累加器 1 的双字逐位右移，空出的位添 0

2．有符号数右移指令

图 3-83 中的 SHR_I 指令将有符号 16 位整数右移 4 位。打开 PLCSIM，将程序下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。在 PLCSIM 中将 −8000 输入 MW40，在变量表中设置 MW40 和 MW42 的显示格式为 BIN（二进制），可以看到右移 4 位的效果（见图 3-84）。右移 4 位相当于除以 2^4，移位后的数为 −500。右移后空出来的位用符号位 1 填充。

令 MW38 分别为 0 和 20（移位位数大于 16），观察移位的结果。移位位数大于 16

时，原有的数据被全部移出去了，MW42 的各位均为符号位 1，其值为 16#FFFF。

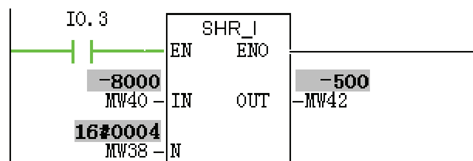


图 3-83　有符号数右移指令

	地址		显示格式	状态值
1	MW	40	BIN	2#1110_0000_1100_0000
2	MW	42	BIN	2#1111_1110_0000_1100

图 3-84　变量表

下面是用 STEP 7 转换图 3-83 中的程序得到的语句表程序。

```
      A     I     0.3
      JNB   _001                  //I0.3 为 0 则跳转
      L     MW    38              //移位次数送累加器 1
      L     MW    40              //移位次数自动送累加器 2，MW40 的值装入累加器 1
      SSI                         //有符号整数右移
      T     MW    42              //移位结果传送到 MW42
_001: NOP   0
```

3．无符号数移位指令

图 3-85 是无符号字左移 4 位的移位指令，在 PLCSIM 中将 50 输入 MW44。在变量表中设置 MW44 和 MW46 的显示格式为 BIN（二进制），可以看到左移 4 位的效果（见图 3-86）。左移 4 位相当于乘以 2^4，移位后的数为 800，左移后空出来的位添 0。

将移位次数分别修改为 2、8 和 16，观察移位的结果。

将图 3-85 中的 OUT 的实参 MW46 改为 MW44，观察程序运行的结果并解释原因。在 I0.4 的触点右边添加一个上升沿检测线圈，用 I0.4 的上升沿启动移位，观察程序运行的结果。

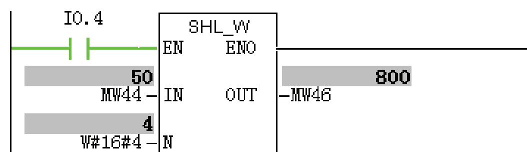


图 3-85　无符号数左移指令

	地址		显示格式	状态值
1	MW	44	BIN	2#0000_0000_0011_0010
2	MW	46	BIN	2#0000_0011_0010_0000

图 3-86　变量表

4．循环移位指令

循环移位指令（见图 3-87）将累加器 1 的整个内容逐位循环左移或循环右移若干位（见表 3-13），即移出来的位又送回累加器 1 另一端空出来的位，最后移出的位装入状态字的 CC1 位。N 为移位的位数，移位的结果保存在输出参数 OUT 指定的地址。

表 3-13　循环移位指令

语句表	梯形图	描述
RLD	ROL_DW	累加器 1 的双字循环左移
RRD	ROR_DW	累加器 1 的双字循环右移
RLDA	—	累加器 1 的双字通过 CC1 循环左移
RRDA	—	累加器 1 的双字通过 CC1 循环右移

在 PLCSIM 中输入 MD50 和 MW48 的十六进制数值，在变量表中设置 MD50 和 MD54 的显示格式为 BIN（二进制），图 3-87 和图 3-88 给出了双字循环左移 8 位的例子。



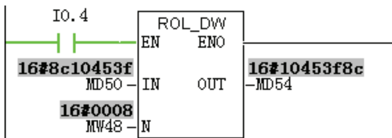


图 3-87　循环左移指令

	地址	显示格式	状态值
1	MD 50	BIN	2#1000_1100_0001_0000_0100_0101_0011_1111
2	MD 54	BIN	2#0001_0000_0100_0101_0011_1111_1000_1100

图 3-88　变量表

将移位次数分别修改为 0、4、16 和 20，观察移位的结果。

3.4.3　实训二十　字逻辑运算指令的仿真实验

1．字逻辑运算指令

字逻辑运算指令（见表 3-14）对两个 16 位字或 32 位双字逐位进行逻辑运算，一个操作数在累加器 1，另一个操作数在累加器 2，或者在指令中用立即数（常数）的形式给出。

如果字逻辑运算的结果为 0，状态字的 CC1 位为 1，反之为 0。状态字的 CC0 和 OV 位被清零。

表 3-14　字逻辑运算指令

语　句　表	梯　形　图	描　　述	语　句　表	梯　形　图	描　　述
AW	WAND_W	字与	AD	WAND_DW	双字与
OW	WOR_W	字或	OD	WOR_DW	双字或
XOW	WXOR_W	字异或	XOD	WXOR_DW	双字异或

“与”运算时两个操作数的同一位如果均为 1，运算结果的对应位为 1，否则为 0（见图 3-89 和图 3-90）。

“或”运算时两个操作数的同一位如果均为 0，运算结果的对应位为 0，否则为 1。

“异或”运算的两个操作数的同一位如果不相同，运算结果的对应位为 1，否则为 0。

2．梯形图中的字逻辑运算指令的仿真实验

OB1 中的字逻辑运算指令见图 3-89，打开 PLCSIM，将程序下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。

生成变量表（见图 3-90），在变量表中输入有关的地址，显示格式均为二进制（BIN）。在“修改值”列设置各输入变量的值，单击工具栏上的按钮，将修改值写入 PLC，观察状态值列各指令的输出变量是否正确。

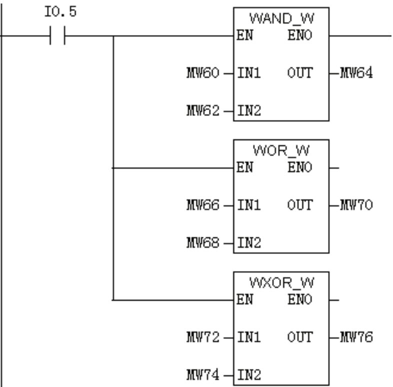


图 3-89　字逻辑运算指令

VAT_1 -- @S7_Pro2\SIMATI...

	地址	显示格式	状态值
1	MW 60	BIN	2#1000_1001_1010_1101
2	MW 62	BIN	2#0010_0011_0111_1011
3	MW 64	BIN	2#0000_0001_0010_1001
4	MW 66	BIN	2#0101_0111_0011_1001
5	MW 68	BIN	2#0010_0110_0111_1000
6	MW 70	BIN	2#0111_0111_0111_1001
7	MW 72	BIN	2#0101_0110_1101_1110
8	MW 74	BIN	2#1000_1000_0110_1011
9	MW 76	BIN	2#1101_1110_1011_0101

图 3-90　变量表

3．字逻辑运算指令的仿真练习

要求将输入 IW0 的高 4 位清零后，保存在 MW0（提示：IW0 与常数 W#16#FFF 作字逻辑“与”运算）。将 QW0 的最高 2 位置 1，其他位保持不变（提示：QW0 与常数 W#16#C000 作字逻辑“或”运算后送 QW0）。编写梯形图程序，下载到仿真 PLC，调试程序，用变量表检查是否满足要求。

3.5 练习题

1．填空

1）M____是 MW100 中的最低位。

2）MB____是 MD100 中最低的 8 位对应的字节。

3）WORD（字）是 16 位___符号数，INT（整数）是 16 位___符号数。

4）Q4.2 是输出字节______的第___位。

5）RLO 是________________的简称。

6）状态字的_____位与方框指令的使能输出 ENO 的状态相同。

7）状态字的_____位与位逻辑指令中的位变量的状态相同。

8）算术运算有溢出或执行了非法的操作时，状态字的_____位被置 1。

9）接通延时定时器的 SD 线圈______时开始定时，定时时间到时剩余时间值为___，其定时器位变为______，其常开触点______，常闭触点_____。定时期间如果 SD 线圈断电，定时器的剩余时间___________。线圈重新通电时，又从_______开始定时。复位输入信号为 1 时，定时器位变为________。定时器位为 1 时如果 SD 线圈断电，定时器的常开触点_____。

10）在加计数器的设置输入 S 端的________，将预置值 PV 指定的值送入计数器字。在加计数脉冲输入信号 CU 的_______，如果计数值小于_____，计数值加 1。复位输入信号 R 为 1 时，计数值被_____。计数值大于 0 时计数器位（即输出 Q）为___；计数值为 0 时，计数器位为___。

11）L#20 是___位的______常数。

12）S5T#10S 和 T#10S 二者之一能用于梯形图的是__________。

2．指出图 3-91 中的错误，左侧垂直线断开处是相邻网络的分界点。

3．怎样打开和关闭梯形图和语句表中的符号显示和符号信息？

4．怎样关闭程序中的注释？怎样才能在打开块时不显示注释？

5．设计立即读取 I3.5 的程序，将读取的值存放在 M3.0 中。

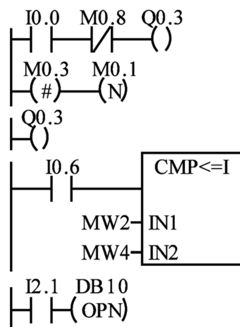


图 3-91 改错

第 4 章

S7-300/400 的用户程序结构

4.1 功能与功能块

4.1.1 S7-300/400 的用户程序结构

1．逻辑块

CPU 循环执行操作系统程序，每次循环都要调用一次主程序 OB1。

STEP 7 将用户编写的程序和程序所需的数据放置在块中，OB、FB、FC、SFB 和 SFC 都是有程序的块，它们称为逻辑块（见表 4-1）。逻辑块类似于子程序，使用户程序结构化，可以简化程序组织，使程序易于修改、查错和调试。程序运行时所需的数据和变量存储在数据块中。

表 4-1 用户程序中的块

块的类型		简要描述
逻辑块	组织块（OB）	操作系统与用户程序的接口，决定用户程序的结构
	功能块（FB）	用户编写的包含经常使用的功能的子程序，有专用的存储区（背景数据块）
	功能（FC）	用户编写的包含经常使用的功能的子程序，没有专用的存储区
	系统功能块（SFB）	集成在 CPU 模块中，通过 SFB 调用系统功能，有专用的存储区（背景数据块）
	系统功能（SFC）	集成在 CPU 模块中，通过 SFC 调用系统功能，没有专用的存储区
数据块	背景数据块（DI）	用于保存 FB 和 SFB 的输入、输出参数和静态变量，其数据是自动生成的
	共享数据块（DB）	存储用户数据的数据区域，供所有的逻辑块共享

系统功能块和系统功能集成在 S7 CPU 的操作系统中，不占用用户程序空间。它们是预先编好程序的逻辑块，可以在用户程序中调用这些块，但是用户不能打开和修改它们。FB 和 SFB 有专用的存储区，其变量保存在指定给它们的背景数据块中。FC 和 SFC 没有背景数据块。

逻辑块可以调用 OB 之外的逻辑块，被调用的块又可以调用别的块，称为嵌套调用。

如果出现中断事件，CPU 将停止当前正在执行的程序，去执行中断事件对应的组织块（即中断程序）。执行完后，返回到程序中断处继续执行。

2．数据块

数据块是用于存放执行用户程序时所需数据的数据区。与逻辑块不同，数据块没有指令，STEP 7 按数据生成的顺序自动地为数据块中的变量分配地址。

4.1.2 实训二十一 功能的生成与调用

1. 生成功能

用新建项目向导生成名为“FC 例程”的项目（见随书光盘中的同名例程），CPU 为 CPU 315-2DP。

执行 SIMATIC 管理器的菜单命令“插入”→“S7 块”→“功能”（见图 4-1），在出现的“属性 – 功能”对话框中，默认的名称为 FC1，设置“创建语言”为 LAD（梯形图）。单击“确定”按钮后，在 SIMATIC 管理器右边窗口出现 FC1。

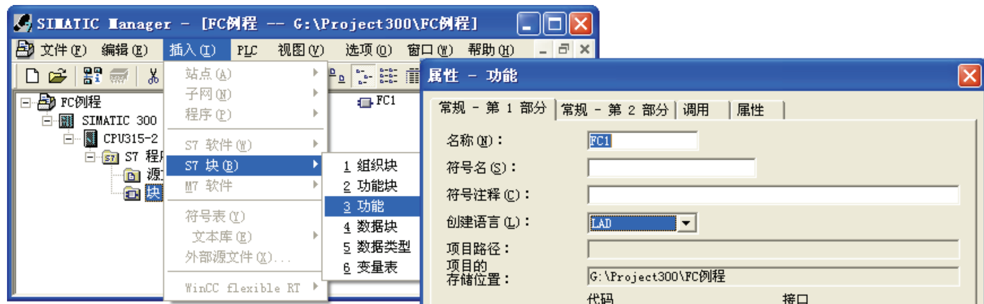


图 4-1 生成功能

2. 生成局部数据

双击打开 FC1（见图 4-2），将鼠标的光标放在程序区最上面的分隔条上，按住鼠标的左键，往下拉动分隔条，分隔条上面是功能的变量声明表，下面是程序区，左边是指令列表和库。将水平分隔条拉至程序编辑器视窗的顶部，不再显示变量声明表，但是它仍然存在。

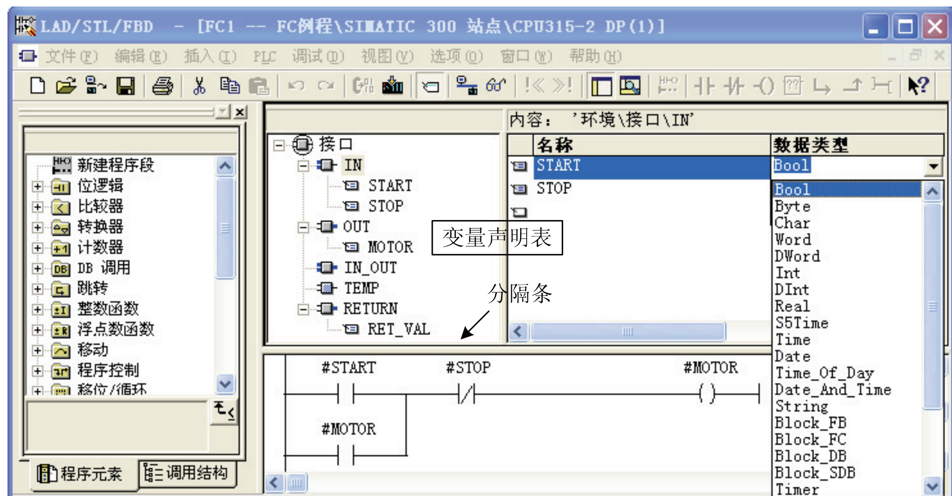


图 4-2 程序编辑器

在变量声明表中声明（即定义）局部变量，局部变量只能在它所在的块中使用。

块的局部变量名必须以英语字母开始，只能由字母、数字和下划线组成，不能使用汉字。

由图 4-2 可知，功能有 5 种局部变量：

1）IN：由调用它的块提供的输入参数。



2）OUT：返回给调用它的块的输出参数。

3）IN_OUT(输入_输出参数)：初值由调用它的块提供，块执行后返回给调用它的块。

4）TEMP (临时数据)：暂时保存在局部数据堆栈中的数据。只是在执行块时使用临时数据，执行完后，不再保存临时数据的数值，它可能被别的数据覆盖。

5）RETURN 中的 RET_VAL（返回值），属于输出参数。

选中变量声明表左边窗口中的“IN”，在变量声明表的右边窗口输入参数的名称 START（起动按钮），按〈Enter〉（回车）键后，自动生成数据类型 Bool（二进制的位）。该参数的下面出现空白行，输入第二个 Bool 型的参数 STOP（停止按钮）。

用同样的方法，生成 Bool 型的输出参数 MOTOR（电动机）。

单击某个数据的“数据类型”列，再单击该单元左边出现的▼按钮，可以选用打开的数据类型列表中的数据类型（见图 4-2）。

在变量声明表中赋值时，不需要指定存储器地址；根据各变量的数据类型，程序编辑器自动地为所有局部变量指定存储器地址。

3．生成功能中的程序

在变量声明表下面的程序区生成梯形图程序（见图 4-2），STEP 7 自动地在局部变量的前面添加#号，例如“#START”。

4．调用功能的仿真实验

双击打开 SIMATIC 管理器中的 OB1，打开程序编辑器左边窗口中的文件夹 FC 块，将其中的 FC1 拖放到右边的程序区的“导线”上。FC1 的方框中左边的 START 等是在 FC1 的变量声明表中定义的输入参数，右边的 MOTOR 是输出参数。它们被称为 FC 的形式参数，简称为形参，形参在 FC 内部的程序中使用。别的逻辑块调用 FC 时，需要为每个形参指定实际的参数（简称为实参），例如，为形参 START 指定的实参为 I0.0（见图 4-3）。



图 4-3　在 OB1 中调用功能

打开 PLCSIM，将所有的逻辑块下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。打开 OB1，单击工具栏上的按钮，起动程序状态监控功能（见图 4-3）。

单击 PLCSIM 中 I0.0 对应的小方框，模拟按下起动按钮。图 4-3 中 I0.0 的值变为 1。I0.0 的状态变化传递给 FC1 的形参 START，如果监控 FC1 内部的程序（见图 4-4），可以看到因为 START 的常开触点闭合，使 MOTOR 的线圈通电。它的值返回给它对应的实参 Q4.0，图 4-3 中 Q4.0 的值变为 1。再单击一次，令 I0.0 为 0 状态，模拟放开起动按钮。

单击两次 PLCSIM 中 I0.1 对应的小方框，模拟按下和放开停止按钮。由于 FC1 中程序的作用，FC1 的输出参数 MOTOR 和它的实参 Q4.0 的值变为 0 状态。

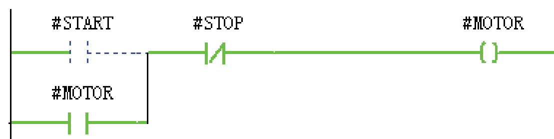


图 4-4　FC1 的程序状态

5．功能的返回值

FC1 的局部变量表中的返回值 RET_VAL 是自动生成的，可以看到它没有初始的数据类型。在调用 FC1 时，方框内没有 RET_VAL。在变量声明表中将它设置为任意的数据类型，在其他逻辑块中调用 FC1 时，可以看到 FC1 方框内右边出现了形参 RET_VAL。由此可知 RET_VAL 属于 FC 的输出参数。

6．功能的仿真练习

设计求圆周长的功能 FC2，FC2 的输入参数为直径 Diameter（INT 整数），圆周率为 3.14159，用整数运算指令计算圆的周长，存放在双字输出参数 Perimeter 中。TMP1 是 FC2 中的双字临时局部变量。在 OB1 中调用 FC2，直径的输入值为常数 10000，存放圆周长的地址为 MD8。

打开 PLCSIM，将所有的逻辑块下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。打开 OB1，单击工具栏上的按钮，起动程序状态监控功能。观察 MD8 中的运算结果是否正确。

4.1.3　实训二十二　功能块的生成与调用

1．生成功能块

功能块是用户编写的有自己的存储区（背景数据块）的逻辑块，功能块的输入、输出参数和静态变量（STAT）用指定的背景数据块 (DI) 存放，临时变量存储在局部数据堆栈中。功能块执行完后，背景数据块中的数据不会丢失，但是不会保存它的临时变量。

调用功能块和系统功能块时需要为它们指定一个背景数据块，后者随功能块的调用而打开，在调用结束时自动关闭。

用新建项目向导生成一个名为“FB 例程”的项目（见随书光盘中的同名例程），CPU 为 CPU 315-2DP。执行 SIMATIC 管理器的菜单命令“插入”→“S7 块”→“功能块”，在出现的“属性 – 功能块”对话框中（见图 4-5），默认的名称为 FB1，将创建语言设置为 LAD（梯形图）。单击“多情景标题”（有的版本为多重背景）复选框，去掉其中的☑，取消多重背景功能。单击“确定”按钮后，在 SIMATIC 管理器右边窗口出现 FB1。

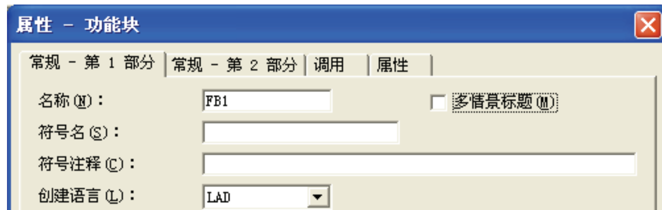


图 4-5　FB1 的属性对话框

2．生成局部变量

控制要求如下：用输入参数“Start”（起动按钮）和“Stop”（停止按钮）控制输出参



数“Motor”（电动机）。按下停止按钮，输入参数 TOF 指定的断电延时定时器开始定时，输出参数“Brake”（制动器）为 1 状态，经过设置的时间预置值后，停止制动。图 4-6 的上面是 FB1 的变量声明表，下面是程序。

输入参数 Speed（实际转速）与静态变量 PreSpeed（预置转速）比较，实际转速大于预置转速时，输出参数 OverSpeed（转速过高，Bool 变量）为 1 状态。

块的输入、输出参数的数据类型可以使用基本数据类型、复杂数据类型、Timer（定时器）、Counter（计数器）、块（FB、FC、DB）、Pointer（指针）、ANY 等。

本例程的输入参数 TOF 的数据类型为 Timer，实参应为定时器的编号（例如 T1）。

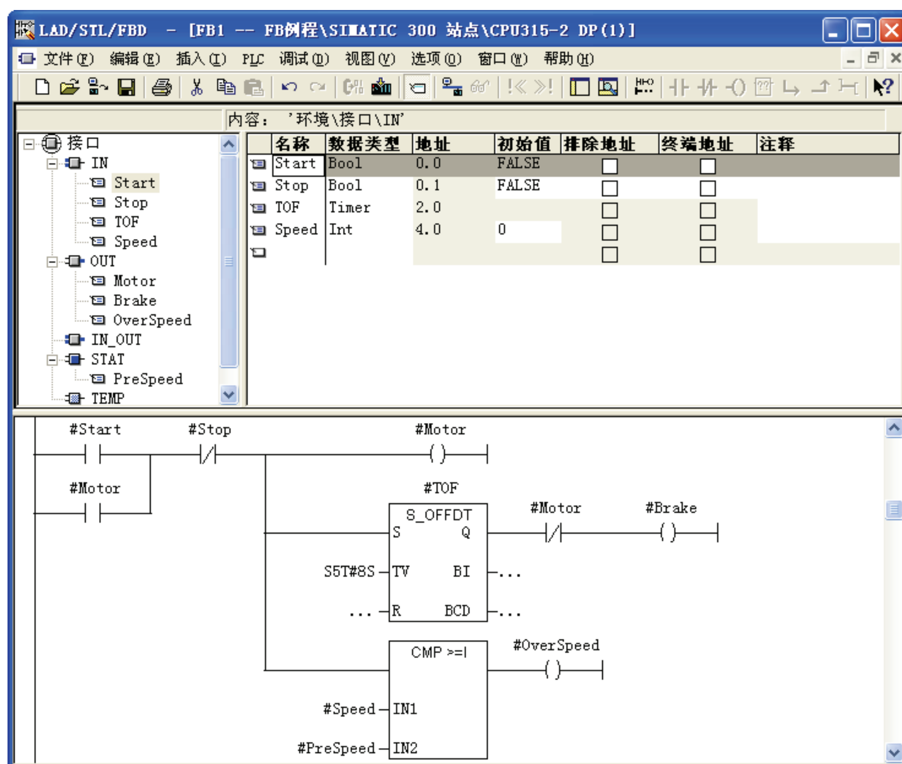


图 4-6　FB1 的局部变量表与程序

从功能块执行完，到下一次重新调用它，其静态变量（STAT）的值保持不变。

3. 在 OB1 中调用 FB1

双击打开 OB1，执行菜单命令“视图”→“总览”，显示出左边的指令列表。打开 FB 文件夹，将其中的 FB1 拖放到程序区的水平“导线”上（见图 4-7）。双击方框上面的红色“???”，输入背景数据块的名称 DB1，按回车键后出现的对话框询问“实例数据块 DB1 不存在，是否要生成它？”。单击“是”按钮确认，打开 SIMATIC 管理器，可以看到自动生成的 DB1。

也可以首先生成 FB1 的背景数据块（见图 4-8），然后在调用 FB1 时使用它。应设置生成的数据块为背景数据块，如果有多个功能块，还应设置是哪一个功能块的背景数据块。

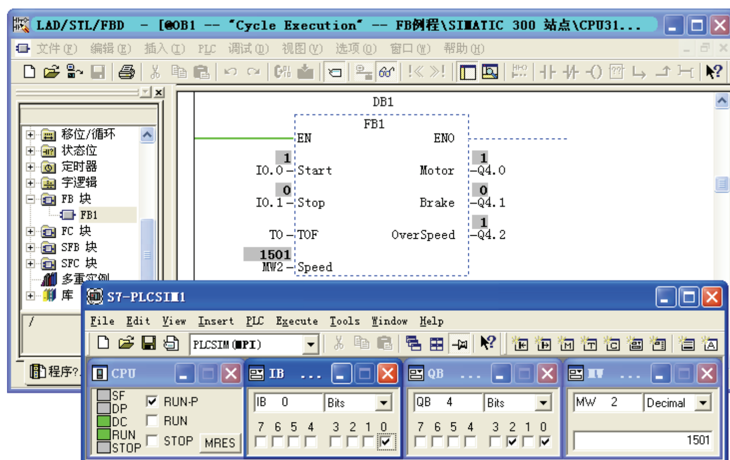


图 4-7　OB1 调用 FB1 的程序状态

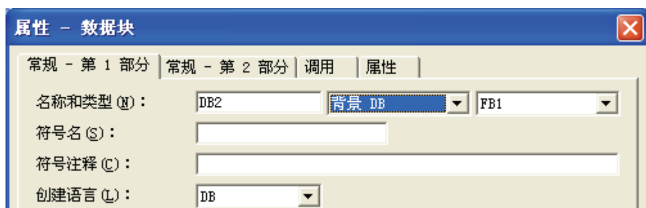


图 4-8　背景数据块的属性对话框

4．背景数据块

背景数据块中的变量就是其功能块的局部变量中的 IN、OUT、IN_OUT 和 STAT 变量（见图 4-6 和图 4-9）。功能块的数据永久性地保存在它的背景数据块中，功能块执行完后也不会丢失，以供下次执行时使用。其他代码块可以访问背景数据块中的变量。不能直接删除和修改背景数据块中的变量，只能在它的功能块的变量申明表中删除和修改这些变量。

生成功能块的输入、输出参数和静态变量时，它们被自动指定一个初始值，可以修改这些初始值。它们被传送给 FB 的背景数据块，作为同一个变量的初始值。调用 FB 时没有指定实参的形参使用背景数据块中的初始值。

5．仿真实验

打开 PLCSIM，将所有的块下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。打开 OB1，单击工具栏上的 按钮，起动程序状态监控功能（见图 4-7）。

DB 参数 - [DB1 -- FB例程\SIMATIC 300 站点\CPU...

数据块(A)　编辑(E)　PLC(P)　调试(D)　查看(V)　窗口(W)　帮助(H)

	地址	声明	名称	类型	初始值	实际值	备注
1	0.0	in	Start	BOOL	FALSE	FALSE	
2	0.1	in	Stop	BOOL	FALSE	FALSE	
3	2.0	in	TOF	TIMER	T 0	T 0	
4	4.0	in	Speed	INT	0	0	
5	6.0	out	Motor	BOOL	FALSE	FALSE	
6	6.1	out	Brake	BOOL	FALSE	FALSE	
7	6.2	out	OverSpeed	BOOL	FALSE	FALSE	
8	8.0	stat	PreSpeed	INT	1500	1500	

图 4-9　FB1 的背景数据块 DB1



单击两次 PLCSIM 中 I0.0 对应的小方框，模拟按下和放开起动按钮。可以看到 OB1 中 I0.0 的值的变化。由于内部程序的作用，输出参数 Motor 的实参 Q4.0 变为 1 状态。

用 PLCSIM 修改实际转速 MW2 的值，它大于等于转速预置值 PreSpeed 的初始值 1500 时，输出参数 OverSpeed 和它的实参 Q4.2 为 1 状态，反之为 0 状态。

单击两次 I0.1 对应的小方框，模拟按下和放开停止按钮。可以看到 Q4.0 变为 0 状态，电动机停机。同时控制制动的 Q4.1 变为 1 状态，经过程序设置的延时时间后，Q4.1 变为 0 状态。

6．功能块的仿真练习

在项目“FB 例程”的 OB1 中，再调用一次 FB1，背景数据块为 DB2，注意两次调用时 FB1 的实参的地址不能重叠。

打开 PLCSIM，将所有的块下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。打开 OB1，单击工具栏上的按钮，起动程序状态监控功能（见图 4-7）。

分别改变两次调用 FB1 的输入参数，观察输出参数的变化是否符合程序的要求。

7．功能与功能块的区别

FB 和 FC 均为用户编写的子程序，局部变量表中均有 IN、OUT、IN_OUT 和 TEMP 变量。FC 的返回值 Ret_Val 实际上属于输出参数。下面是 FC 和 FB 的区别：

1）功能块有背景数据块，功能没有背景数据块。

2）只能在功能内部访问它的局部变量。其他逻辑块可以访问功能块的背景数据块中的变量。

3）功能没有静态变量（STAT），功能块有保存在背景数据块中的静态变量。

功能如果有执行完后需要保存的数据，只能存放在全局变量（例如全局数据块和 M 区）中，但是这样会影响功能的可移植性。如果功能或功能块的内部不使用全局变量，只使用局部变量，不需要做任何修改，就可以将它们移植到其他项目。如果块的内部使用了全局变量，在移植时需要考虑每个块使用的全局变量是否会与别的块产生地址冲突。

4）功能块的局部变量（不包括 TEMP）有初始值，功能的局部变量没有初始值。在调用功能块时如果没有设置某些输入、输出参数的实参，进入 RUN 模式时将使用背景数据块中的初始值。调用功能时应给所有的形参指定实参。

8．组织块与 FB 和 FC 的区别

1）事件或故障发生时，由操作系统调用对应的组织块，FB 和 FC 是用户程序在逻辑块中调用的。

2）组织块没有输入参数、输出参数和静态参数，只有临时局部变量。组织块自动生成的 20B 临时局部变量包含了与触发组织块的事件有关的信息，它们由操作系统提供。

9．时间标记冲突与一致性检查

调用 FB1 后，在 FB1 的变量声明表中生成一个新的输入参数，保存后关闭 FB1。

双击打开 OB1，出现的小对话框显示“至少一个块调用有时间标志冲突”。单击“帮助”按钮可以获取解决问题的帮助信息。单击“确定”按钮后打开 OB1，可以看到 FB1 的方框和形参、实参均为红色。关闭 OB1，返回 SIMATIC 管理器。选中左边窗口的“块”，执行菜单命令“编辑”→“检查块的一致性”。

在“检查块的一致性”视图（见图 4-10 左边的图），可以看到 OB1 和 DB1 左边红色的故障指示灯。单击工具栏最左边的编译按钮，出现要求关闭程序编辑器的“编译”对话

框（见图 4-10 上面的小图）。单击“确定”按钮，开始编译。编译结束后，红色的故障指示灯全部消失（见图 4-10 右下角的图），下面的编译信息显示没有错误和警告。打开 OB1，可以看到 FB1 上的红色消失，方框左边出现新增加的输入参数。

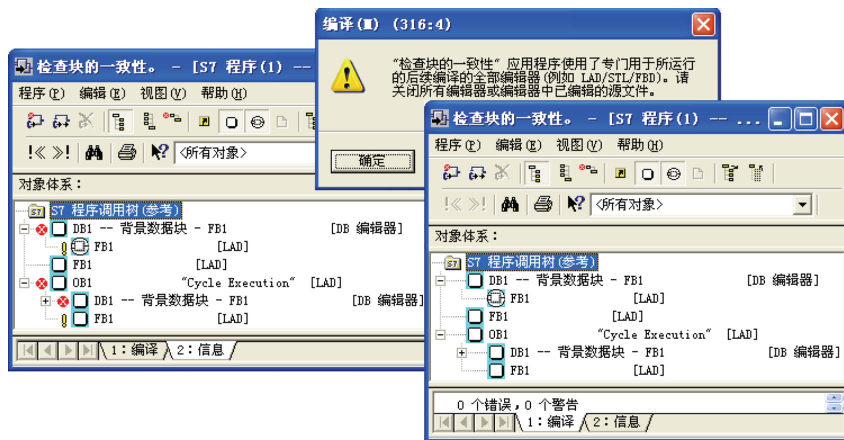


图 4-10　检查块的一致性

如果用上述方法不能消除时间标志冲突，只能删除有冲突的块，然后重新调用修改参数后的块。

4.1.4　实训二十三　共享数据块与系统功能的调用

1．生成数据块

数据块（DB）用来分类储存设备或生产线中变量的值，分为共享数据块和背景数据块。共享数据块和符号表中的变量可供所有的逻辑块使用。

CPU 可以用 OPN 指令分别打开一个共享数据块和一个背景数据块。打开数据块 DB1 后，DB1.DBW2 可以简写为 DBW2。打开新的数据块时，原来被打开的数据块自动关闭。

用新建项目向导生成一个名为“数组_SFC”的项目（见随书光盘中的同名例程），CPU 为 CPU 315-2DP。

执行 SIMATIC 管理器的菜单命令“插入”→“S7 块”→“数据块”命令（见图 4-1），生成一个名为 DB1 的数据块，默认的类型为共享数据块（见图 4-11）。

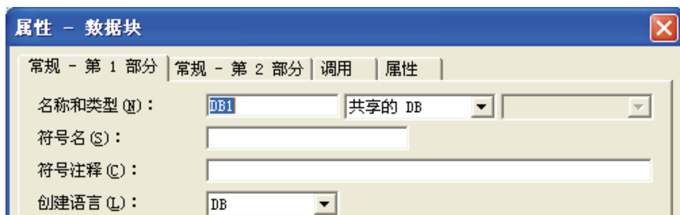


图 4-11　数据块的属性对话框

2．生成数组

数据块的大小与数据块中变量的个数和数据类型有关。数组由同一类型的数据组合而成，可以用数组快速定义数据块的大小。



双击打开生成的数据块，其中只有一个临时占位符变量 DB_VAR。将变量的名称改为 Press，用鼠标右键单击“类型”列，执行出现的快捷菜单中的“复杂类型”→“ARRAY”（见图 4-12），生成一个数组。



图 4-12　设置数据类型

在出现的“ARRAY[]”的方括号中，输入“1..4”（见图 4-13）。ARRAY[0..4]中的 1 和 4 分别是数组元素的下标的下限值和上限值，它们可以是任意的整数(-32 768~32 767)，上限值应大于下限值。选中注释列的单元后按计算机的回车键，ARRAY[0..4]下面出现空白单元，在其中输入数组元素的数据类型 INT，结束了对数组的声明。

图 4-13 中生成的是一维数组，该数组有 4 个数据类型为 INT 的元素。“初始值”列可以输入用英语的逗号分隔的各元素的初始值。其中的“2（0）”表示最后两个元素的初始值为 0。如果不输入初始值，所有元素的初始值均为默认值 0。此外还生成了一个 INT 变量 Voltage。

图 4-13 的“地址”列中的“+0.0”表示数组的起始字节地址为 0，“*2.0”表示数组元素的大小为 2B，“+8.0”表示数组的大小为 8B。DB1.Press[2]是该数组中下标为 2 的元素。

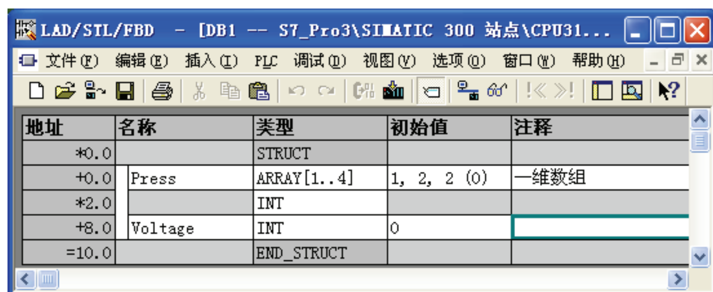


图 4-13　数据块中的数组

3. 调用系统功能

打开 OB1，执行菜单命令“视图”→“总览”，左边窗口出现指令列表。打开最下面的“\库\Standard Library\System Function Blocks”文件夹，可以看到系统功能块 SFB 和系统功能 SFC（见图 4-14）。

SFC21（FILL）用源存储区的内容初始化目标存储区。选中该文件夹中的 SFC21，按计算机的〈F1〉键，打开在线帮助，可以看到对 SFC21 的详细说明和应用实例。

SFC21 的 3 个参数均可以采用 I、Q、M、D（数据块）和 L（局部变量）区。变量 BVAL（源区域）和 BLK（目标区域）的数据类型均为参数类型 ANY。

将上述库文件夹中的 SFC21“FILL”拖放到程序区（见图 4-14），本例调用 SFC21 来将 MW2 的数据传送到 DB1 的数组 Press 的各单元。在执行 SFC 时如果出错，返回值 RET_VAL 中是错误代码。

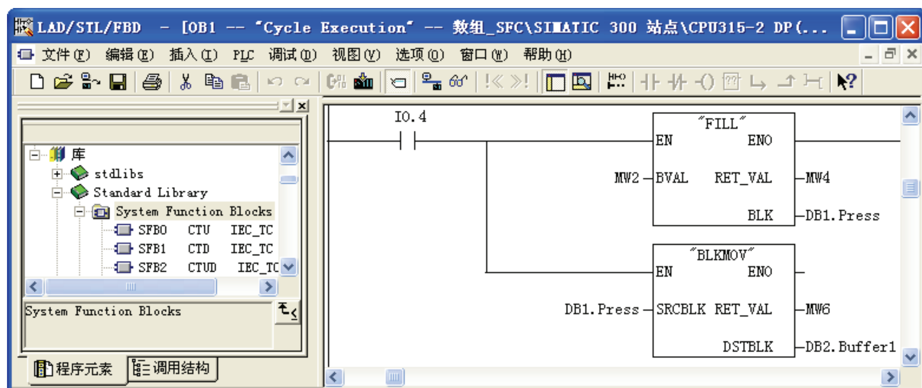


图 4-14　程序编辑器

SFC20“BLKMOV”（块传送）用于将源存储区 SRCBLK 的内容复制到目标存储区 DSTBLK，两个存储区的地址不能重叠。

在 SIMATIC 管理器中生成数据块 DB2，删除自动生成的占位符变量，生成一个有 4 个 INT 元素的数组 Buffer1。在 OB1 中调用 SFC20，将数组 DB1.Press 各元素的值传送给 DB2 中的数组 Buffer1（见图 4-14）。

下面是用语句表调用 SFC20 的程序：

```
CALL   "BLKMOV"                        //调用 SFC20
 SRCBLK    :=DB1.Press                 //源存储器区
 RET_VAL   :=MW10                      //执行 SFC20 出错时的错误代码
 DSTBLK    := DB2.Buffer1              //目标存储器区
```

“:=”号之前是 SFC 的形式参数（形参），在“:=”号的后面输入各形参的实际参数（实参），“//”号的右边是对该行语句的注释。

4．仿真实验

打开 PLCSIM，将逻辑块下载到仿真 PLC，将它由 STOP 模式切换到 RUN-P 模式。打开 DB1，执行菜单命令“视图”→“数据视图”。将数据块切换到数据视图显示方式（见图 4-15），可以看到数组中的各元素。图 4-13 是声明视图显示方式，用于声明（定义）数据块中的数据。

单击工具栏上的“监视（开/关）”按钮，起动监控功能。如果原来数据块是声明视图显示方式，将会自动切换到数据视图显示方法。

图 4-15 中的“初始值”列是生成数组时设置的初始值。在 PLCSIM 中设置 MW2 为 31524，令 I0.4 为 1 状态，OB1 中的 SFC21 被执行。MW2 中的数据 31524 被写入 DB1 中的数组 Press 的各元素。图 4-15 中各数组元素的实际值变为 31524。

双击打开 DB2，单击工具栏上的按钮，起动监控功能。由于 OB1 中 SFC20 的作用，DB2 中的数组元素的值均为 31524。



5．数据块应用练习

在符号名为 Pump 的数据块中生成一个由 50 字节组成的一维数组（下标从 1 到 50），数组的符号名为 Press。用数据视图方式观察数组中的各元素。



图 4-15 数据视图显示方式的数组与 PLCSIM

4.1.5 实训二十四 多重背景的应用

1．多重背景的概念

有时需要多次调用同一个功能块来控制同一类型的被控对象，每次调用都需要一个背景数据块，但是这些背景数据块中的变量又很少，这样在项目中就出现了大量的背景数据块“碎片”。在用户程序中使用多重背景数据块可以减少背景数据块的数量。

多重背景用得不是太多，初学者可以暂时不学。

例程“多重背景”用项目“FB 例程”中的 FB1 来控制两台电动机。如果在 OB1 中调用两次 FB1，需要使用两个背景数据块 DB1 和 DB2。使用多重背景时只需要一个背景数据块，但是需要增加一个功能块 FB10 来调用两次作为“局部背景”的 FB1，FB1 的数据存储在 FB10 的背景数据块 DB10 中。不需要给 FB1 分配背景数据块，即原来的 DB1 和 DB2 被 DB10 代替，但是需要在 FB10 的变量声明表中声明数据类型为 FB1 的两个静态变量（STAT）。

2．多重背景功能块

用新建项目向导生成一个名为“多重背景”的项目（见随书光盘中的同名例程），CPU 为 CPU 315-2DP。首先生成与项目“FB 例程”相同的 FB1。

执行 SIMATIC 管理器的菜单命令“插入”→“S7 块”→“功能块”，在出现的“属性 – 功能块”对话框中（见图 4-16），设置块的名称为 FB10，将创建语言设置为 LAD（梯形图）。采用默认的设置，激活“多情景标题”（多重背景）复选框。单击“确定”按钮后，在 SIMATIC 管理器右边窗口出现 FB10。

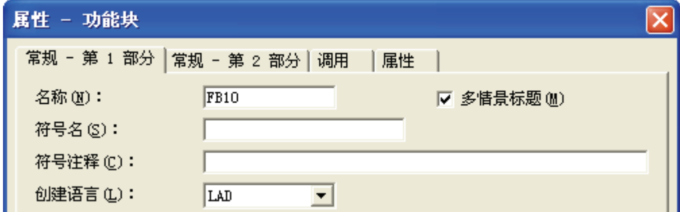


图 4-16 生成多重背景数据块

在 FB10 的变量声明表中（见图 4-17），声明了名为“Motor1”和“Motor2”的静态变量（STAT），其数据类型为 FB1。变量声明表的文件夹“Motor1”和“Motor2”内的 8 个变量与 FB1 的 8 个局部变量相同，它们是自动生成的。

完成上述操作后，“Motor1”和“Motor2”将出现在程序编辑器左边目录窗口的“多重实例”（即多重背景）文件夹内（见图 4-17）。将它们“拖放”到 FB10 的程序区中，然后指定它们的输入参数和输出参数。

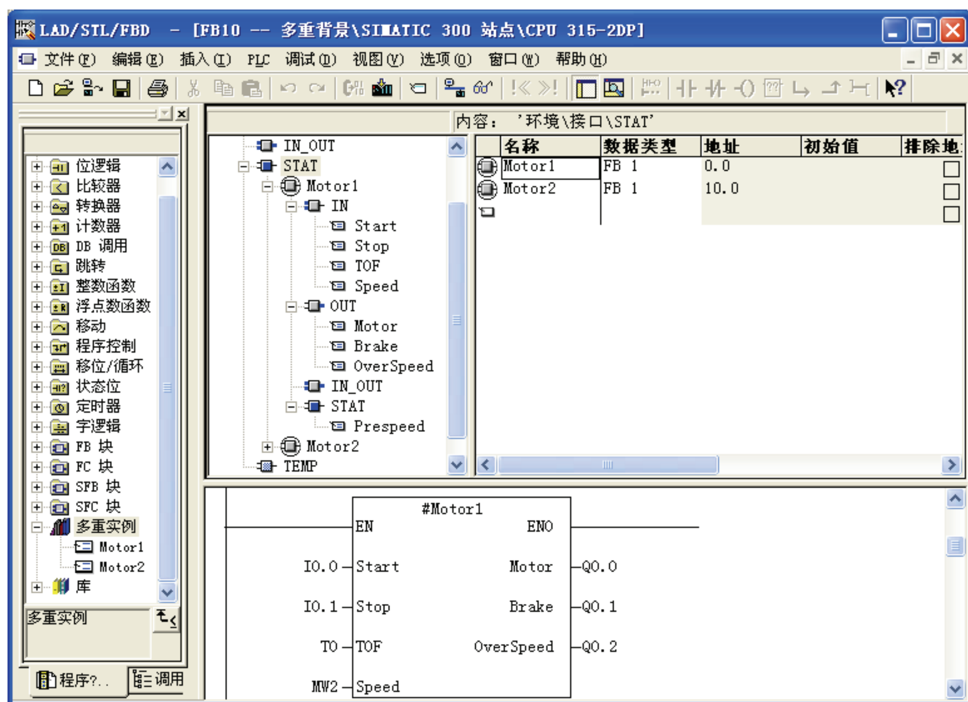


图 4-17 定义与调用多重背景

在 OB1 中调用 FB10，其背景数据块为 DB10（见图 4-18）。在本例中，FB10 没有输入参数和输出参数。

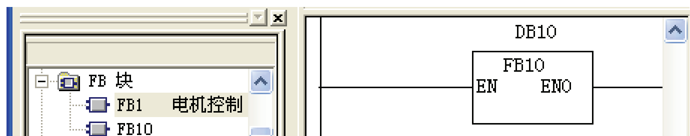


图 4-18 OB1 调用 FB10 的程序

控制两台电动机的局部变量均存储在多重背景数据块 DB10 中（见图 4-19），DB10 的变量是自动生成的，与 FB10 的变量声明表中的相同（不包括临时变量）。

3．仿真实验

打开 PLCSIM，将所有的逻辑块下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。打开 FB10，单击工具栏上的按钮，起动程序状态监控功能。



DB 参数 - [DB10 -- 多重背景\SIMATIC 300 站点\CPU315-2

	地址	声明	名称	类型	初始值	实际值	备注
1	0.0	stat:in	Motor1.Start	BOOL	FALSE	FALSE	
2	0.1	stat:in	Motor1.Stop	BOOL	FALSE	FALSE	
3	2.0	stat:in	Motor1.TOF	TIMER	T 0	T 0	
4	4.0	stat:in	Motor1.Speed	INT	0	0	
5	6.0	stat:out	Motor1.Motor	BOOL	FALSE	FALSE	
6	6.1	stat:out	Motor1.Brake	BOOL	FALSE	FALSE	
7	6.2	stat:out	Motor1.OverSpeed	BOOL	FALSE	FALSE	
8	8.0	stat	Motor1.PreSpeed	INT	1500	1500	
9	10.0	stat:in	Motor2.Start	BOOL	FALSE	FALSE	
10	10.1	stat:in	Motor2.Stop	BOOL	FALSE	FALSE	
11	12.0	stat:in	Motor2.TOF	TIMER	T 0	T 0	
12	14.0	stat:in	Motor2.Speed	INT	0	0	
13	16.0	stat:out	Motor2.Motor	BOOL	FALSE	FALSE	
14	16.1	stat:out	Motor2.Brake	BOOL	FALSE	FALSE	
15	16.2	stat:out	Motor2.OverSpeed	BOOL	FALSE	FALSE	
16	18.0	stat	Motor2.PreSpeed	INT	1500	1500	

图 4-19 多重背景数据块

调试程序的方法与项目“FB 例程”相同，分别改变两台设备的输入参数，观察输出参数的变化是否符合程序的要求。图 4-20 是调试 2 号设备时的程序状态监视画面和 PLCSIM。

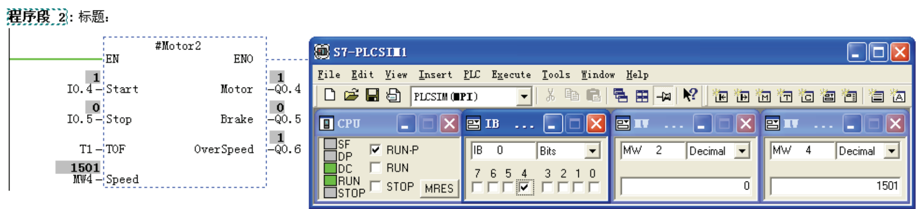


图 4-20 多重背景的程序状态监视

4.2 组织块的应用

4.2.1 组织块与中断

1. 组织块

组织块（见表 4-2）是操作系统与用户程序的接口，由操作系统调用，组织块中的程序是用户编写的。组织块用于控制扫描循环和中断程序的执行、PLC 的启动和错误处理等，可以使用的组织块与 CPU 的型号有关。

表 4-2 组 织 块

OB 编号	错 误 类 型	优 先 级	说 明
OB1	启动或上一次循环结束时执行 OB1	1	主程序循环
OB10~OB17	时间中断 0～7	2	在设置的日期和时间起动
OB20~OB23	时间延迟中断 0～3	3~6	延时后起动
OB30~OB38	循环中断 0～8	7~15	以设定的时间为周期运行
OB40~OB47	硬件中断 0～7	16~23	检测到来自外部模块的中断请求时起动
OB55	状态中断	2	DPV1 中断（PROFIBUS-DP 中断）

（续）

OB 编号	错 误 类 型	优 先 级	说 明
OB56	刷新中断	2	
OB57	制造厂商特殊中断	2	
OB60	多处理器中断，调用 SFC35 时起动	25	多处理器中断的同步操作
OB61~64	同步循环中断 1～4	25	
OB65	技术功能同步中断	25	
OB70	I/O 冗余错误	25	冗余故障中断，只用于 H 系列 CPU
OB72	CPU 冗余错误	28	
OB73	通信冗余错误	25	
OB80	时间错误	26、28（对于 S7-300） 25、28（对于 S7-400 和 CPU 318）	异步错误中断
OB81	电源故障		
OB82	诊断中断		
OB83	插入/取出模块中断		
OB84	CPU 硬件故障		
OB85	优先级错误		
OB86	机架故障或分布式 I/O 的站故障		
OB87	通信错误		
OB90	背景组织块	29	背景循环
OB100~OB102	暖启动、热启动、冷启动	27	启动
OB121	编程错误	与引起错误的 OB 的优先级相同	同步错误中断
OB122	I/O 访问错误		

OB1 用于循环处理，是用户程序中的主程序。操作系统在每一次循环中调用一次 OB1。

2．事件中断处理

中断处理用来实现对特殊内部事件或外部事件的快速响应。如果没有中断事件发生，CPU 循环执行组织块 OB1。CPU 检测到中断源的中断请求时，操作系统在执行完当前逻辑块的当前指令后，立即响应中断，自动调用中断源对应的中断组织块。执行完中断组织块后，返回被中断的程序的断点处继续执行原来的程序。中断组织块不是由逻辑块调用，而是在中断事件发生时由操作系统调用。中断组织块中的程序是用户编写的。

有中断事件发生时，如果没有下载对应的组织块，CPU 将会进入 STOP 模式。如果用户希望忽略某个中断事件，可以生成和下载一个对应的空的组织块，出现该中断事件时，CPU 不会进入 STOP 模式。

3．中断的优先级

OB 按触发事件分成几个级别，这些级别有不同的优先级（见表 4-2）。如果在执行中断程序（组织块）时，又检测到一个中断请求，CPU 将比较两个中断源的中断优先级。如果优先级相同，按照产生中断请求的先后次序进行处理。如果后者的优先级比正在执行的 OB 的优先级高，将中止当前正在处理的 OB，改为调用较高优先级的 OB。这种处理方式称为中断程序的嵌套调用。



4．组织块的临时局部变量

每个组织块的局部数据区都有 20B 的临时变量（TEMP），它们提供触发该 OB 的事件的详细信息，这些信息在 OB 启动时由操作系统提供（见表 4-3）。

表 4-3　OB 的临时局部变量

地址（字节）	内　　容
0	事件级别与标识符，例如 OB40 为 B#16#11，表示硬件中断被激活
1	用代码表示与启动 OB 的事件有关的信息
2	优先级，例如 OB40 的优先级为 16
3	OB 块号，例如 OB40 的块号为 40
4～11	事件的附加信息，例如 OB40 的 LB5 为产生中断的模块的类型，LW6 为产生中断的模块的起始地址；LD8 为产生中断的通道号
12～19	OB 被启动的日期和时间（年、月、日、时、分、秒、毫秒与星期）

4.2.2　实训二十五　使用循环中断的彩灯控制程序

本节的实训主要用于熟悉启动组织块与循环中断组织块的使用方法。

1．CPU 模块的启动方式与启动组织块

打开 CPU 模块的属性对话框的“启动”选项卡，S7-400 可以选择暖启动、热启动和冷启动这 3 种启动方式中的一种，绝大多数 S7-300 CPU 只能暖启动。

OB100～OB102 是启动组织块，用于系统初始化。CPU 上电或运行模式由 STOP 切换到 RUN 时，CPU 只是在第一个扫描循环周期执行一次启动组织块。

1）暖启动：过程映像数据以及非保持的存储器位、定时器和计数器被复位。具有保持功能的存储器位、定时器、计数器和所有的数据块将保留原数值。执行一次 OB100 后，循环执行 OB1。

2）热启动：如果 S7-400 CPU 在 RUN 模式时电源突然丢失，然后又很快重新上电，将执行 OB101，自动地完成热启动，从上次 RUN 模式结束时程序被中断之处继续执行，不对计数器等复位。

3）冷启动：上述的所有系统存储区均被清除，即被复位为零，包括有保持功能的存储区。用户程序从装载存储器载入工作存储器，调用 OB102 后，循环执行 OB1。

用户可以通过在启动组织块 OB100～OB102 中编写程序，来设置 CPU 的初始化操作，例如设置开始运行时某些变量的初始值和输出模块的初始值等。

2．循环中断组织块

循环中断组织块用于按精确的时间间隔循环执行中断程序，例如周期性地执行闭环控制系统的 PID 控制程序，间隔时间从 STOP 切换到 RUN 模式时开始计算。

时间间隔不能小于 5ms。如果时间间隔过短，还没有执行完循环中断程序又开始调用它，将会产生时间错误事件，调用 OB80。如果没有创建和下载 OB80，CPU 将进入 STOP 模式。

大多数 S7-300 CPU 只能使用 OB35，其余的 CPU 可以使用的循环中断 OB 的个数与

CPU 的型号有关。

3．硬件组态

用新建项目向导生成名为“OB35 例程”的项目（见随书光盘中的同名例程），CPU 为 CPU 315-2DP。双击硬件组态工具 HW Config 中的 CPU，打开 CPU 属性对话框（见图 4-21），由“周期性中断”选项卡可知只能使用 OB35，其循环周期的默认值为 100ms，将它修改为 1000ms，将组态数据下载到 CPU 后生效。如果没有下载，循环周期为默认值 100ms。



图 4-21　组态循环中断

相位偏移量（默认值为 0）用于错开 S7-400 不同时间间隔的几个循环中断 OB，使它们不会被同时执行，以减少连续执行循环中断 OB 的时间。

单击工具栏上的按钮，编译并保存组态信息。

4．OB100 的程序

用鼠标右键单击 SIMATIC 管理器左边窗口中的“块”，在弹出的菜单中执行“插入新对象”→“组织块”命令，在出现的“属性 – 组织块”对话框中（见图 4-22），将组织块的名称改为 OB100，设置创建语言为 LAD（梯形图）。单击“确定”按钮后，在 SIMATIC 管理器右边窗口出现 OB100。

双击打开 OB100（见图 4-23），用 MOVE 指令将 MB0 的初值置为 7，即低 3 位置 1，其余各位为 0。此外用 ADD_I 指令将 MW6 加 1，可以观察 CPU 执行 OB100 的次数。

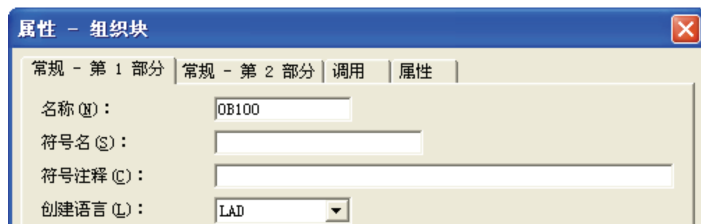


图 4-22　OB100 的属性对话框



图 4-23　OB100 的程序



5. OB35 的程序

OB35 中的程序用于控制 8 位彩灯循环移位，用 I0.0 控制移位的方向。I0.0 为 1 状态时彩灯左移，为 0 状态时彩灯右移。

S7-300/400 只有双字循环移位指令，MB0 是双字 MD0 的最高字节（见图 4-24）。在 MD0 每次循环左移 1 位之后，最高位 M0.7 的数据被移到 MD0 最低位的 M3.0。为了实现 MB0 的循环移位，移位后如果 M3.0 为 1 状态，将 MB0 的最低位 M0.0 置位为 1（见图 4-26 的程序段 1），反之将 M0.0 复位为 0，相当于 MB0 的最高位 M0.7 移到了 MB0 的最低位 M0.0。

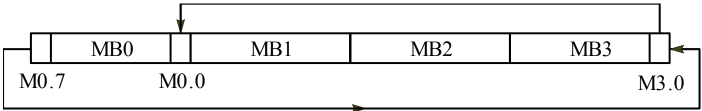


图 4-24　MB0 循环左移

在 MB0 每次循环右移 1 位之后（见图 4-25），MB0 的最低位 M0.0 的数据被移到 MB1 最高位的 M1.7。移位后根据 M1.7 的状态，将 MB0 的最高位 M0.7 置位或复位（见图 4-26 的程序段 2），相当于 MB0 的最低位 M0.0 移到了 MB0 的最高位 M0.7。

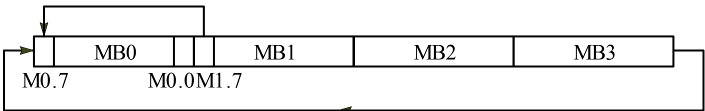


图 4-25　MB0 循环右移

在程序段 3，用 MOVE 指令将 MB0 的值传送到 QB4，用 QB4 来控制 8 位彩灯。

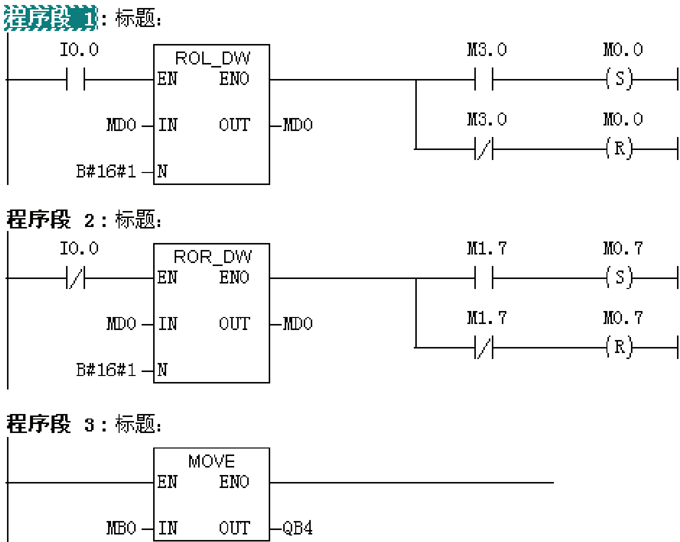


图 4-26　OB35 的程序

6. 禁止和激活硬件中断

SFC40“EN_IRT”和 SFC39“DIS_IRT”分别用于激活和禁止中断和异步错误的系统功

能。它们的参数 MODE 为 2 时激活指定的 OB 编号对应的中断，MODE 必须用十六进制数来设置。OB_NR 是中断的编号。

在 OB1 中编写图 4-27 的程序，在 I0.2 的上升沿调用 SFC“EN_IRT”激活 OB35 对应的循环中断，在 I0.3 的上升沿时调用 SFC“DIS_IRT”禁止 OB35 对应的循环中断。

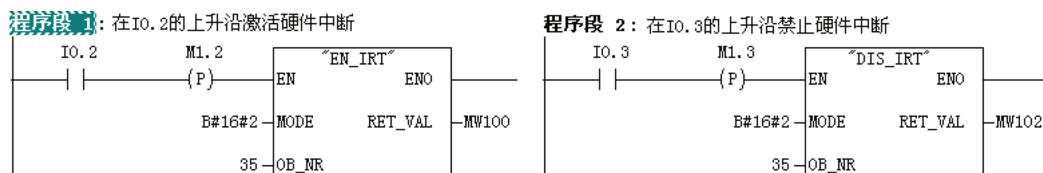


图 4-27　OB1 激活和禁止循环中断的程序

7．仿真实验

打开仿真软件 PLCSIM，下载系统数据和所有的块后，切换到 RUN-P 模式，CPU 调用一次 OB100，MW6 被加 1（见图 4-28），说明只调用了一次 OB35。MB0 被设置为初始值 7，其低 3 位为 1。OB35 被自动激活，CPU 每 1s 调用一次 OB35。因为 I0.0 的初始值为 0，QB4 的值每 1s 循环右移 1 位。

将 I0.0 置为 1 状态，QB4 由循环右移变为循环左移。

单击两次 I0.3 对应的小方框，在 I0.3 的上升沿，循环中断被禁止，CPU 不再调用 OB35，QB0 的值固定不变。单击两次 I0.2 对应的小方框，在 I0.2 的上升沿，循环中断被激活，QB0 的值又开始循环移位。

改变 OB100 中 MB0 的初始值后，下载到仿真 PLC，观察运行的效果。

8．仿真练习

要求每 500ms 调用一次 OB35，每次调用时将 MW30 加 1。编写程序后下载到仿真 PLC，调试程序直到满足要求。

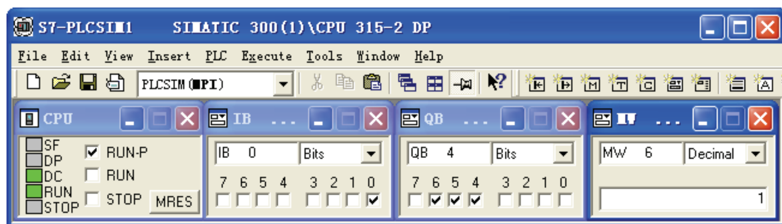


图 4-28　PLCSIM

4.2.3　实训二十六　时间中断组织块的应用

可以设置在某一特定的日期时间产生一次时间中断，也可以设置从设定的日期时间开始，周期性地重复产生中断，例如每分钟、每小时、每天、每周、每月、甚至每年产生一次时间中断。可以用专用的 SFC28～SFC30 设置、取消和激活时间中断。绝大多数 S7-300 CPU 只能使用 OB10。

1．基于硬件组态的时间中断

要求在到达设置的日期和时间时，用 Q4.0 自动起动某台设备。用新建项目向导生成一个



名为“OB10_1”的项目（见随书光盘中的同名例程），CPU 模块的型号为 CPU 315-2DP。

打开硬件组态工具 HW Config（见图 4-29），双击机架中的 CPU，打开 CPU 的属性对话框。在“时刻中断”选项卡，设置执行起动设备的日期和时间，执行的方式为“一次”。用复选框激活中断，按“确定”按钮结束设置。单击工具栏上的按钮，保存和编译组态信息。

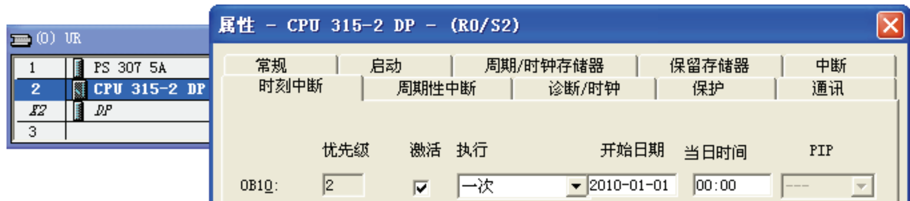


图 4-29　组态时间中断

在 SIMATIC 管理器中生成 OB10，下面是用语句表编写的 OB10 的程序，设置的时间到时，将需要起动的设备对应的输出点置位：

```
SET                    //将 RLO 置为 1
=      Q     4.0       //将 RLO 写入 Q4.0
```

下面是 OB1 中的程序，用 I0.0 将 Q4.0 复位：

```
A      I     0.0
R      Q     4.0
```

单击工具栏上的按钮，打开 PLCSIM，生成 QB4 的视图对象。下载所有的块和系统数据后，将仿真 PLC 切换到 RUN-P 模式。达到设置的日期和时间时，可以看到 Q4.0 变为 1 状态。

做实验时设置比当前的日期时间稍晚一点的日期和时间，以免等待时间太长。

2. 用 SFC 控制时间中断

除了在硬件组态工具中设置和激活时间中断之外，也可以在用户程序中用 SFC28“SET_TINT”和 SFC30“ACT_TINT”设置和激活时间中断。

用新建项目向导生成一个名为“OB10_2”的项目（见随书光盘中的同名例程）。

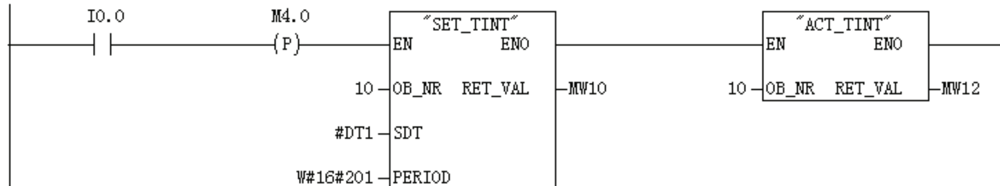
在 OB1 中调用 SFC31“QRY_TINT”来查询时间中断的状态（见图 4-30），读取的状态字在 MW8 中。

IEC 功能 FC3“D_TOD_TD”用于合并日期和时间值，它在程序编辑器左边窗口的文件夹“\库\Standard Library\IEC Function Blocks”中。生成 OB1 的临时局部变量（TEMP）“DT1”，其数据类型为 Date_And_Time，FC3 的执行结果保存在 DT1 中。

在 I0.0 的上升沿，调用 SFC28“SET_TINT”和 SFC30“ACT_TINT”来分别设置和激活时间中断 OB10。在 I0.1 的上升沿，调用 SFC29“CAN_TINT”来禁止时间中断。

各 SFC 的参数 OB_NR 是组织块编号，SFC28 的参数 SDT 是开始产生中断的日期和时间。PERIOD 用来设置执行的方式，W#16#0201 表示每分钟产生一次时间中断。RET_VAL 是执行时可能出现的错误代码，为 0 时无错误。





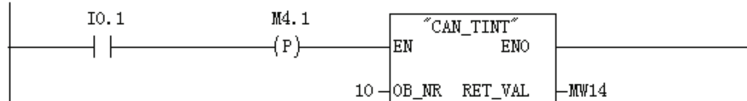


图 4-30　OB1 的程序

下面是 OB10 中将 MW2 加 1 的 STL 程序：

```
L    MW    2
+    1
T    MW    2
```

3．仿真实验

打开仿真软件 PLCSIM，生成 MB9、IB0 和 MW2（见图 4-31）的视图对象，MB9 是 SFC31 读取的状态字 MW8 的低位字节。下载所有的块后，将仿真 PLC 切换到 RUN-P 模式，M9.4 变为 1 状态，表示已经下载了 OB10。

令 I0.0 为 1 状态，M9.2 变为 1 状态，表示时间中断已被激活，CPU 每分钟调用一次 OB10，将 MW2 加 1。两次单击 I0.1 对应的小方框，在 I0.1 的上升沿，时间中断被禁止，M9.2 变为 0 状态，MW2 停止加 1。两次单击 I0.0 对应的小方框，在 I0.0 的上升沿，时间中断被重新激活，M9.2 变为 1 状态，MW2 每分钟又被加 1。

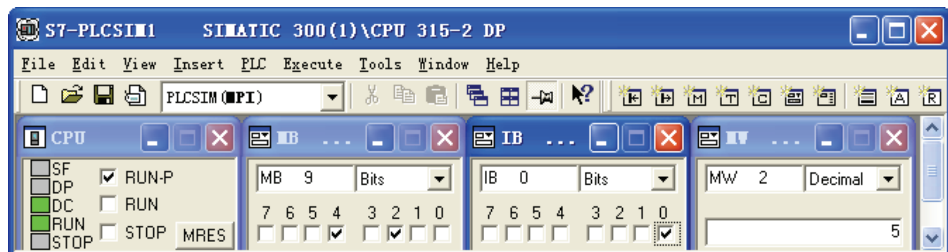


图 4-31　PLCSIM

4.2.4　实训二十七　硬件中断组织块的应用

硬件中断组织块（OB40～OB47）用于快速响应信号模块（SM，即输入/输出模块）、



通信处理器（CP）和功能模块（FM）的信号变化。具有中断能力的上述模块将中断信号传送到 CPU 时，将触发硬件中断。绝大多数 S7-300 CPU 只能使用 OB40。S7-400 CPU 可以使用的硬件中断 OB 的个数与 CPU 的型号有关。

产生硬件中断时，如果没有生成和下载硬件中断组织块，操作系统将会向诊断缓冲区输入错误信息，并执行异步错误处理组织块 OB80。

1．硬件组态

用新建项目向导生成一个名为“OB40 例程”的项目（见随书光盘中的同名例程），CPU 模块的型号为 CPU 315-2DP。

选中 SIMATIC 管理器左边的 300 站对象，双击右边窗口的“硬件”图标，打开硬件组态工具 HW Config（见图 4-32）。将硬件目录中名为“DI4xNAMUR，Ex”的 4 点 DI 模块插入 4 号槽，16 点 DO 模块插入 5 号槽。

自动分配的 DI 模块的字节地址为 0。双击该模块，打开它的属性对话框（见图 4-32）。用复选框启用硬件中断，设置 I0.0 产生上升沿中断，I0.1 产生下降沿中断。



图 4-32　组态硬件中断

2．编写 OB40 中的程序

OB40 中的程序（见图 4-33）判断是哪个模块的哪个点产生的中断，然后执行相应的操作。临时局部变量 OB40_MDL_ADDR 和 OB40_POINT_ADDR 分别是产生中断的模块的起始字节地址和模块内的位地址，数据类型分别为 Word 和 DWord，这两个变量不能直接用于整数比较指令和双整数比较指令。将它们保存到 MW6 和 MD8 后，用比较指令判别是哪一个模块和哪一点产生的中断。在 I0.0 的上升沿将 Q4.0 置位，在 I0.1 的下降沿将 Q4.0 复位。

3．硬件中断的仿真实验

打开 PLCSIM（见图 4-34），下载所有的块后，将仿真 PLC 切换到 RUN-P 模式。执行 PLCSIM 的菜单命令“Execute”（执行）→“Trigger Error OB”（触发错误 OB）→“Hardware Interrupt (OB40-OB47)…”，打开“Hardware Interrupt OB（40-47）”对话框（见图 4-34 右上方的小图），在文本框“Module address”（模块地址）内输入模块的起始地址 0，在文本框“Module status（POINT_ADDR）”（模块状态（位地址））内输入模块内的位地址 0。

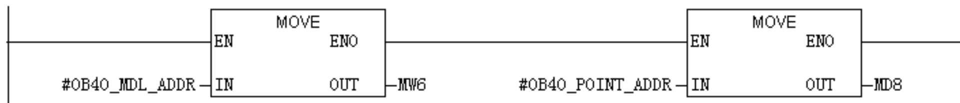






图 4-33　OB40 中的程序

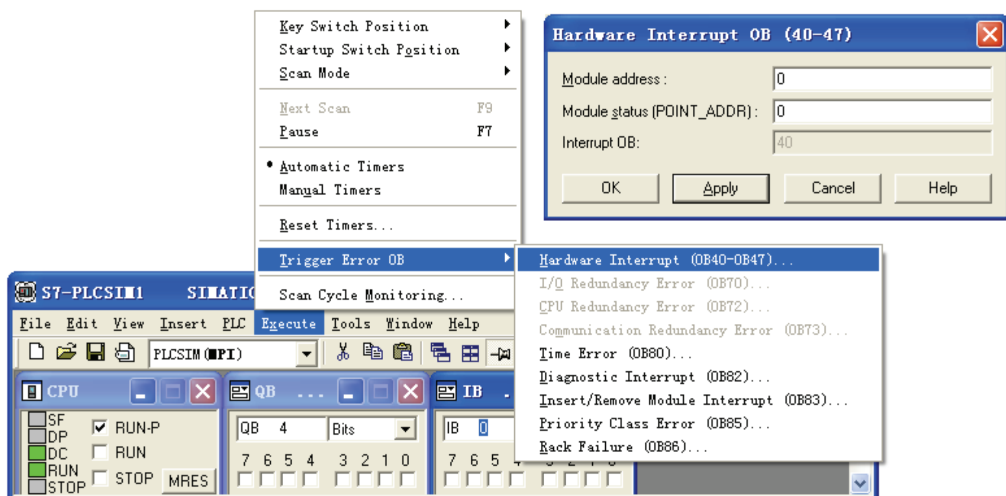


图 4-34　Execute 菜单中的命令和模拟硬件中断的对话框

单击“Apply”（应用）按钮，触发设置的 I0.0 的上升沿中断，CPU 调用 OB40，Q4.0 被置为 1 状态，同时在“Interrupt OB”（中断 OB）显示框内自动显示出对应的 OB 编号 40。将位地址 POINT_ADDR 改为 1，模拟 I0.1 产生的中断，单击“Apply”按钮，在放开按钮时，Q4.0 被复位为 0 状态。单击“OK”按钮，将执行与“Apply”同样的操作，同时关闭对话框。

4．禁止和激活硬件中断

在 OB1 中编写图 4-35 中的程序，在 I0.2 的上升沿调用 SFC40（EN_IRT）激活 OB40 对应的硬件中断，在 I0.3 的上升沿调用 SFC39（DIS_IRT）禁止 OB40 对应的硬件中断。SFC 中的 MODE 为 2 时用 OB 的编号 OB_NR 指定中断。

点击两次 PLCSIM 中 I0.3 对应的小方框，OB40 被禁止执行。这时用图 4-34 右上角的



对话框模拟产生硬件中断，不会调用 OB40。点击两次 I0.2 对应的小方框，OB40 被允许执行。又可以用 I0.0 和 I0.1 来控制 Q4.0 了。

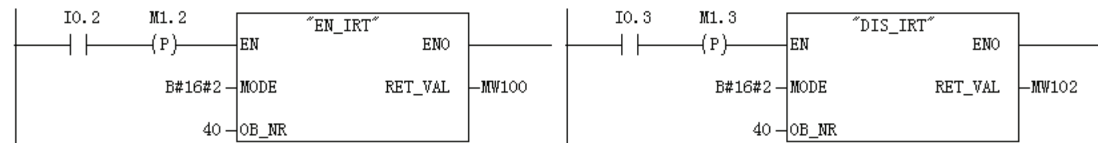


图 4-35　OB1 中激活和禁止硬件中断的程序

4.2.5　实训二十八　延时中断组织块的应用

PLC 的普通定时器的定时精度较差，如果需要高精度的延时，可以使用延时中断 OB。

用 SFC32“SRT_DINT”启动延时中断，延迟时间为 1～60000ms，精度为 1ms。延时时间到时触发中断，调用 SFC32 指定的 OB。CPU 316 及以下的 CPU 只能使用 OB20。

1．硬件组态

用新建项目向导生成一个名为“OB20”的项目（见随书光盘中的同名例程），CPU 模块的型号为 CPU 315-2DP。

打开硬件组态工具 HW Config，将硬件目录中名为“DI4xNAMUR，Ex”的 4 点 DI 模块插入 4 号槽，自动分配的 DI 模块的字节地址为 0。双击该模块，打开它的属性对话框（见图 4-32）。用复选框启用硬件中断，设置 I0.0 产生上升沿中断。

2．程序设计

在 OB1 中调用 SFC34“QRY_DINT”来查询延时中断的状态字 STATUS（见图 4-36），查询的结果保存在 MW8，其低字节为 MB9。OB_NR 是延时中断 OB 的编号，RET_VAL 为 SFC 执行时的错误代码，为 0 时无错误。

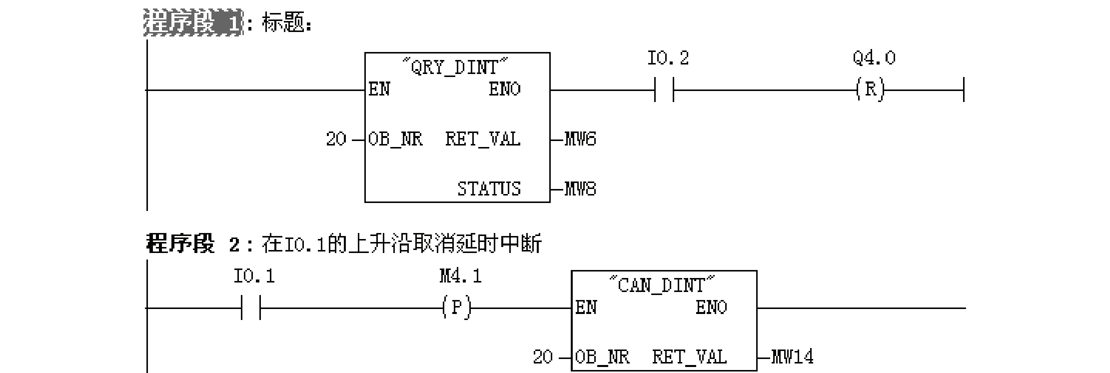


图 4-36　OB1 中的程序

在延时过程中，可以用 I0.1 调用 SFC33“CAN_DINT”来取消延时中断过程。

在 I0.0 的上升沿触发硬件中断，CPU 调用 OB40，在 OB40 中调用 SFC32“SRT_DINT”启动延时中断（见图 4-37），延时时间为 10s。从 LD12 开始的 8B 临时局部变量是调用 OB40 的日期时间值，用 MOVE 指令保存其中的后 4 个字节 LD16。

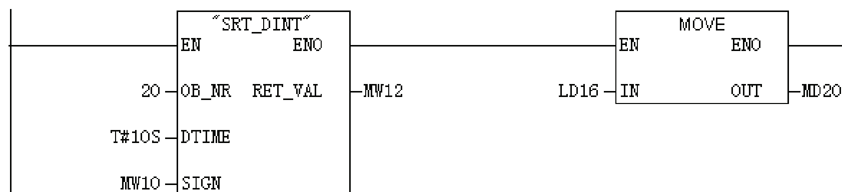


图 4-37　OB40 中的程序

10s 后延时时间到，CPU 调用 SFC32 指定的 OB20。在 OB20 中用 MOVE 指令保存调用 OB20 的日期时间值的后 4 个字节（见图 4-38）。同时将 Q4.0 置位，并通过 PQB4 立即输出。可以用 I0.2 将 Q4.0 复位（见图 4-36）。

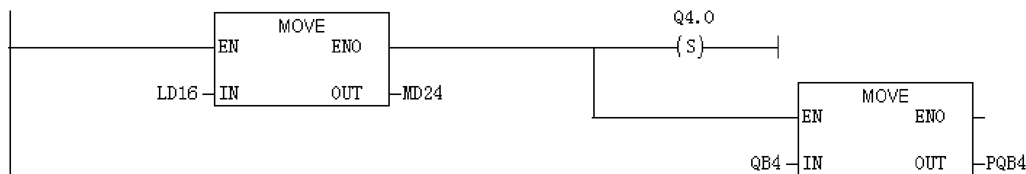


图 4-38　OB20 中的程序

3．仿真实验

打开仿真软件 PLCSIM，将程序和组态信息下载到仿真 PLC。切换到 RUN-P 模式时，M9.4 马上变为 1 状态（见图 4-39），表示 OB20 已经下载到了 CPU 中。

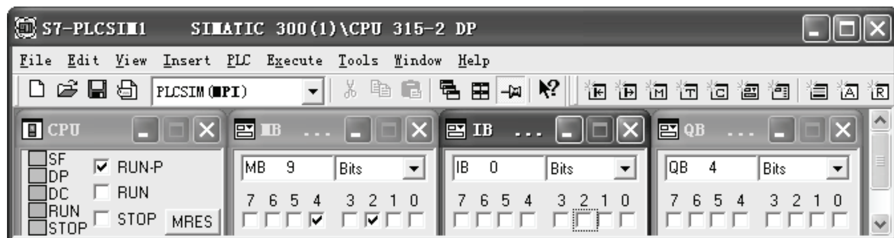


图 4-39　PLCSIM

执行 PLCSIM 的菜单命令“Execute”→“Trigger Error OB”→“Hardware Interrupt (OB40-OB47)…”（见图 4-34），在“Hardware Interrupt OB（40-47）”对话框中，输入模块的起始字节地址 0 和模块内的位地址 0。单击“Apply”按钮，I0.0 产生硬件中断，CPU 调用 OB40，M9.2 变为 1 状态，表示正在执行 SFC 32 启动的时间延时。

在 SIMATIC 管理器中生成变量表（见图 4-40），单击工具栏上的按钮，起动监控功能。MD20 是在 OB40 中读取的 BCD 格式的时间值（25 分 9 秒 643 毫秒），最后 1 位为星期的代码，5 表示星期 4。

10s 的延时时间到时，CPU 调用 OB20，M9.2 变为 0 状态，表示延时结束。OB20 中的程序将 Q4.0 置位为 1 状态（见图 4-38），并且用 MOVE 指令立即写入 DO 模块。可以用 I0.2 复位 Q4.0。在 OB20 中保存在 MD24 的当前时间值为 25 分 19 秒 643 毫秒，与 OB40 中保存的时间值相减，可知定时精度是相当高的。



变量 - [VAT_1 -- @OB20\SIMATIC 300(1)\CPU 315-2 DP\S...

表格(T)　编辑(E)　插入(I)　PLC　变量(A)　视图(V)　选项(O)　窗口(W)　帮助(H)

	地址		符号	显示格式	状态值	修改数值
1	MD	20		HEX	DW#16#25096435	
2	MD	24		HEX	DW#16#25196435	

图 4-40　变量表

在延时过程中用仿真软件将 I0.1 置位为 1，M9.2 变为 0 状态，表示 OB20 的延时被取消，定时时间到不会调用 OB20。

4.3　练习题

1．填空

1）逻辑块包括_____、________、________、____________和_______________。

2）CPU 可以同时打开___个共享数据块和___个背景数据块。用______指令打开 DB2 后，DB2.DBB0 可以用_____来访问。

3）背景数据块中的数据是功能块的______________中的数据（不包括临时数据）。

4）调用_________和______________时需要指定其背景数据块。

5）若 FB1 调用 FC1，应先创建二者中的______。

6）在梯形图中调用功能块时，方框内是功能块的______，方框外是对应的______。方框的左边是块的______参数，右边是块的______参数。

7）S7-300 在起动时调用 OB______。

8）CPU 检测到错误时，如果没有下载对应的错误处理 OB，CPU 将进入_______模式。

2．功能和功能块有什么区别？

3．组织块与其他逻辑块有什么区别？

4．怎样生成多重背景功能块，怎样调用多重背景？

5．延时中断与定时器都可以实现延时，它们有什么区别？

6．在程序中怎样表示 DB1 中的数组 Press 的下标为 15 的元素？

第 5 章

梯形图的顺序控制设计法

5.1 顺序控制设计法

5.1.1 顺序功能图

1．顺序控制设计法

继电器电路图和简单的梯形图程序一般采用经验设计法来设计，这种设计方法具有很大的试探性和随意性，很难掌握。设计的质量和速度与设计者的经验有很大的关系。

所谓顺序控制，就是按照生产工艺预先规定的顺序，在各个输入信号的作用下，根据内部状态和时间的顺序，在生产过程中各个执行机构自动地有秩序地进行操作。

顺序控制设计法是一种先进的很容易掌握的设计方法，对于有经验的工程师，也会提高设计的效率，程序的调试、修改和阅读也很方便。

顺序功能图（Sequential Function Chart，SFC）是描述控制系统的控制过程、功能和特性的一种图形，也是 PLC 的编程语言标准 IEC 61131-3 位居首位的编程语言。S7-300/400 的 S7 GRAPH 就是一种顺序功能图语言。

现在还有相当多的 PLC（包括 S7-200 和 S7-1200）没有配备顺序功能图语言。可以用顺序功能图来描述系统的功能，根据它来设计梯形图程序。本章首先介绍顺序功能图的画法，然后介绍用置位复位指令设计顺序控制程序的方法，最后介绍用顺序功能图语言 S7 GRAPH 设计顺序控制程序的方法。

2．步的基本概念

顺序控制设计法最基本的思想是将系统的一个工作周期划分为若干个顺序相连的阶段，这些阶段称为步（Step），并用编程元件（例如位存储器 M）来代表各步。步是根据输出量的状态变化来划分的，在任何一步之内，各输出量的 ON/OFF 状态不变，但是相邻两步输出量总的状态是不同的，步的这种划分方法使代表各步的编程元件的状态与各输出量的状态之间，有着极为简单的逻辑关系。

顺序控制设计法用转换条件控制代表各步的编程元件，让它们的状态按规定的顺序变化，然后用代表各步的编程元件去控制 PLC 的各输出位。

图 5-1 中的两条运输带的控制要求与 3.2.3 节中的相同。起动时应先启动 1 号运输带，延时 6s 后自动起动 2 号运输带。按了停止按钮后，先停 2 号运输带，5s 后再停 1 号运输带。图 5-1 给出了输入输出信号的波形图和顺序功能图。控制 1 号运输带的 Q4.0 在步 M0.1～M0.3 中都应为 1 状态。



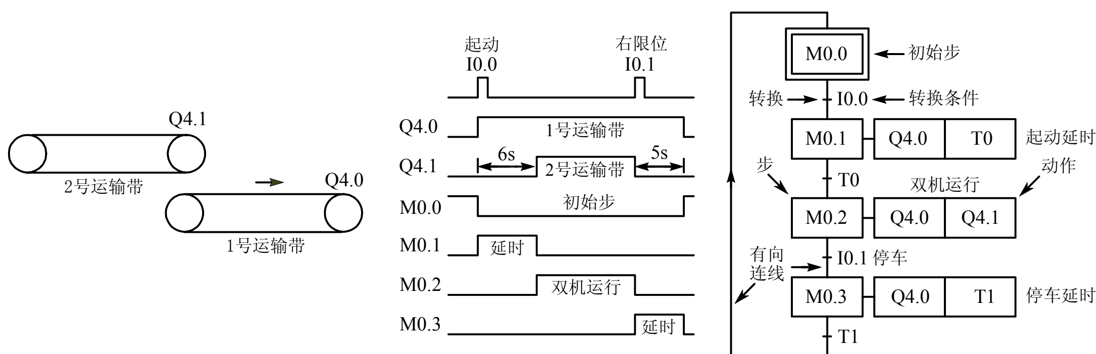


图 5-1　波形图与顺序功能图

根据 Q4.0～Q4.1 的 ON/OFF 状态的变化，显然可以将上述工作过程分为 3 步，分别用 M0.1～M0.3 来代表这 3 步，另外还设置了一个等待起动的初始步 M0.0。图 5-1 的右边是描述该系统的顺序功能图，图中用矩形方框表示步，方框中是代表该步的编程元件的地址，例如 M0.0 等。

与系统的初始状态相对应的步称为初始步，初始状态一般是系统等待起动命令的相对静止的状态。初始步用双线方框表示，每一个顺序功能图至少应该有一个初始步。

当系统正处于某一步所在的阶段时，称该步处于活动状态，该步为“活动步”。

3．与步对应的动作

在顺序功能图中，PLC 的输出被称为“动作”，动作放在矩形框内，并与它所在的步对应的方框相连。

如果某一步有几个动作，可以用图 5-2 中的两种画法来表示，但是并不隐含这些动作之间有先后顺序。应清楚地表明动作是存储型的还是非存储型的。图 5-1 中的 Q4.0 和 Q4.1 均为非存储型动作，例如在步 M0.2 为活动步时，动作 Q4.1 为 1 状态，步 M0.2 为不活动步时，动作 Q4.1 为 0 状态。

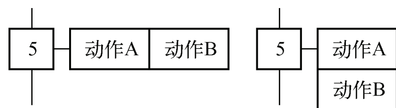


图 5-2　动作

在图 5-1 的步 M0.1 中，接通延时定时器 T0 用于起动延时，在该步 T0 的线圈应一直通电，转换到下一步后，T0 的线圈断电。从这个意义上说，T0 的线圈相当于该步的一个非存储型动作，因此将这种为某一步定时的接通延时定时器放在与该步相连的动作框内，它表示定时器的线圈在该步内“通电”。

4．有向连线

在顺序功能图中，随着时间的推移和转换条件的实现，将会发生步的活动状态的进展，这种进展按有向连线规定的路线和方向进行。在画顺序功能图时，将代表各步的方框按它们成为活动步的先后次序顺序排列，并用有向连线将它们连接起来。步的活动状态默认的进展方向是从上到下或从左至右，在这两个方向有向连线上的箭头可以省略。如果不是上述的方向，则应在有向连线上用箭头注明进展方向。

5．转换与转换条件

转换用有向连线上与有向连线垂直的短划线来表示，转换将相邻两步分隔开。

使系统由当前步进入下一步的信号称为转换条件，转换条件可以是外部的输入信号，例

如按钮、指令开关、限位开关的接通或断开等；也可以是 PLC 内部产生的信号，例如定时器、计数器触点的通断等，转换条件还可以是若干个信号的与、或、非逻辑组合。

图 5-1 中步 M0.1 下面的转换条件 T0 对应于接通延时定时器 T0 的常开触点，在 T0 的定时时间到时该转换条件满足。

6．绘制顺序功能图的练习

冲床的运动示意图如图 5-3 所示。初始状态时机械手在最左边，I0.4 为 1 状态；冲头在最上面，I0.3 为 1 状态；机械手松开，Q0.0 为 0 状态。按下起动按钮 I0.0，Q0.0 变为 1 状态，工件被夹紧并保持，2s 后 Q0.1 变为 1 状态，机械手右行，直到碰到 I0.1。以后将顺序完成以下动作：冲头下行，冲头上行，机械手左行，机械手松开（Q0.0 被复位）。延时 2s 后，系统返回初始状态。各限位开关和定时器提供的信号是相应步之间的转换条件。画出控制系统的顺序功能图。

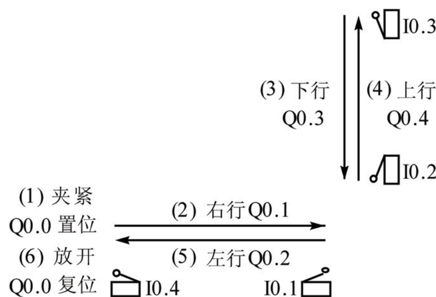


图 5-3　冲床运动示意图

5.1.2　实训二十九　使用置位复位指令的顺序控制程序

1．运输带控制的程序设计

用新建项目向导生成一个名为“运输带顺控”的项目（见随书光盘中的同名例程），CPU 为 CPU 315-2DP。

执行 SIMATIC 管理器的菜单命令“插入”→“S7 块”→“组织块”，将组织块的名称改为“OB100”。单击“确定”按钮确认。

双击打开 OB100，用 MOVE 指令将顺序功能图中的各步（M0.0~M0.3）清零，然后将初始步 M0.0 置位为活动步（见图 5-4）。

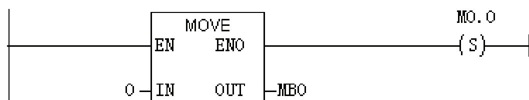


图 5-4　OB100 中的梯形图

2．控制步的转换的电路设计方法

在顺序功能图中，如果某一转换所有的前级步都是活动步，并且满足该转换对应的转换条件，则应实现转换。即该转换所有的后续步都应变为活动步，该转换所有的前级步都应变为不活动步。用该转换所有前级步对应的存储器位（M）的常开触点与转换条件对应的触点或电路串联，来使所有后续步对应的存储器位置位和所有前级步对应的存储器位复位。在任何情况下，代表步的存储器位的控制电路都可以用这一原则来设计，每一个转换对应一个这样的控制置位和复位的电路块，有多少个转换就有多少个这样的电路块。这种设计方法特别有规律，在设计复杂的顺序功能图的梯形图时既容易掌握，又不容易出错。

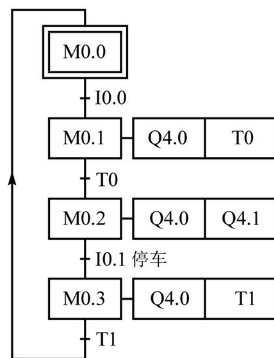


图 5-5　顺序功能图

将图 5-1 的运输带控制系统的顺序功能图重新画在图 5-5 中。



实现初始步下面的 I0.0 对应的转换需要同时满足两个条件，即该转换的前级步是活动步（M0.0 为 1 状态）和转换条件满足（I0.0 为 1 状态）。在梯形图中，用 M0.0 和 I0.0 的常开触点组成的串联电路来表示上述条件。该电路接通时，两个条件同时满足。此时应将该转换的后续步变为活动步，即用置位指令（S 指令）将 M0.1 置位。还应将该转换的前级步变为不活动步，即用复位指令（R 指令）将 M0.0 复位。

图 5-6 中的程序段 1~4 是用上述方法编写的控制步 M0.0～M0.3 的置位复位电路，每一个转换对应一个这样的电路。

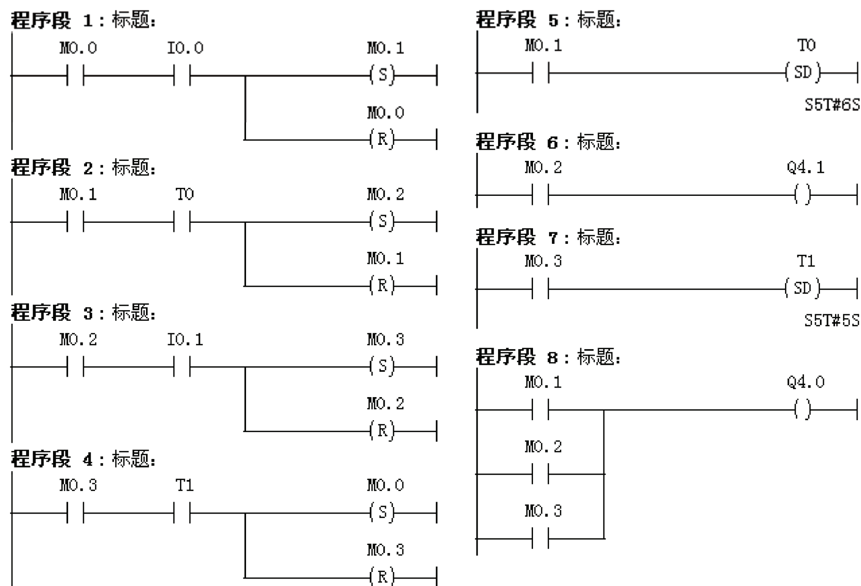


图 5-6　OB1 中的梯形图

3．输出电路的处理

应根据顺序功能图，用代表步的存储器位的常开触点或它们的并联电路来控制输出位的线圈。Q4.1 仅仅在步 M0.2 为 1 状态，它们的波形完全相同（见图 5-1）。因此可以用 M0.2 的常开触点直接控制 Q4.1 的线圈。

接通延时定时器 T0 的线圈仅在步 M0.1 接通，因此用 M0.1 的常开触点控制 T0 的线圈。由于同样的原因，用 M0.3 的常开触点控制 T1 的线圈。

Q4.0 的线圈在步 M0.1~M0.3 均为 1 状态，因此将 M0.1~M0.3 的常开触点并联后，来控制 Q4.0 的线圈。

4．程序的调试

顺序功能图是用来描述控制系统的外部性能的，因此应根据顺序功能图而不是梯形图来调试顺序控制程序。

打开 PLCSIM，生成 IB0、QB4、MB0、T0 和 T1 的视图对象（见图 5-7）。将所有的逻辑块下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。由于执行了 OB100 的程序，初始步对应的 M0.0 为 1 状态，其余各步对应的存储器位为 0 状态。

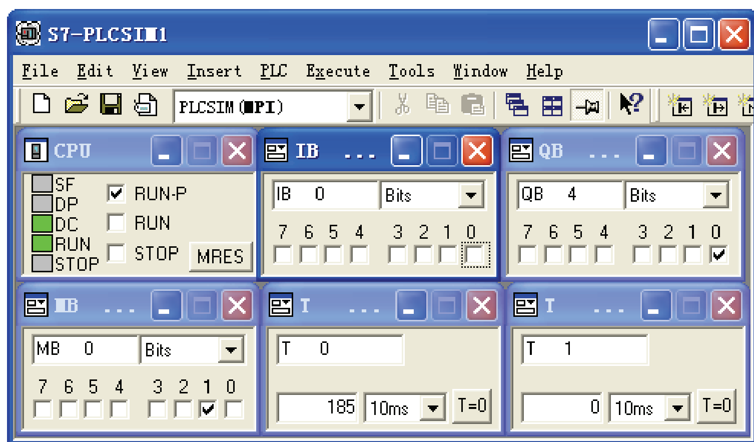


图 5-7 PLCSIM

单击两次 PLCSIM 中 I0.0 对应的小方框，模拟按下和放开起动按钮。初始步下面的转换条件满足，M0.0 变为 0 状态，M0.1 和 Q4.0 变为 1 状态，说明转换到了起动延时步。T0 的当前值从 600（单位为 10ms）开始不断减少（见图 5-7），减到 0 时，6s 延时结束，M0.1 变为 0 状态，M0.2 和 Q4.1 变为 1 状态，说明转换到了步 M0.2。

单击两次 I0.1 对应的小方框，模拟按下和放开停车按钮。步 M0.2 下面的转换条件满足，M0.2 和 Q4.1 变为 0 状态，M0.3 变为 1 状态，说明转换到了停车延时步。T1 的当前值从 500（单位为 10ms）开始不断减少，减到 0 时，5s 的延时结束，M0.3 和 Q4.0 变为 0 状态，M0.0 变为 1 状态，返回到初始步 M0.0。

5．3 条运输带的控制

3 条运输带顺序相连（见图 3-36），按下起动按钮 I0.2，1 号运输带开始运行，5s 后 2 号运输带自动起动，再过 5s 后 3 号运输带自动起动。停机的顺序与起动的顺序刚好相反，即按了停止按钮 I0.3 后，先停 3 号运输带，5s 后停 2 号运输带，再过 5s 停 1 号运输带。Q4.2～Q4.4 分别控制 1~3 号运输带。

画出 3 条运输带控制的顺序功能图，根据顺序功能图，设计出梯形图程序。打开 PLCSIM，将程序下载到仿真 PLC，将 CPU 切换到 RUN-P 模式后调试程序，直到满足要求。

5.1.3 实训三十 专用钻床的顺序控制

1．顺序功能图的基本结构

（1）单序列

单序列由一系列相继激活的步组成，每一步的后面仅有一个转换，每一个转换的后面只有一个步（见图 5-8a），单序列的特点是没有下述的分支与合并。

（2）选择序列

选择序列的开始称为分支（见图 5-8b），转换符号只能标在水平连线之下。如果步 5 是活动步，并且转换条件 h 为 1 状态，则发生由步 5→步 8 的进展。如果步 5 是活动步，并且 k 为 1 状态，则发生由步 5→步 10 的进展。如果将转换条件 k 改为 $k \cdot \overline{h}$，在 k 和 h 同时为 1



状态时，将优先选择 h 对应的序列，一般只允许同时选择一个序列。

选择序列的结束称为合并（见图 5-8b），转换符号只允许标在水平连线之上。

如果步 9 是活动步，并且转换条件 j 为 1 状态，则发生由步 9→步 12 的进展。如果步 11 是活动步，并且 n 为 1 状态，则发生由步 11→步 12 的进展。

（3）并行序列

并行序列用来表示系统的几个同时工作的独立部分的工作情况。并行序列的开始称为分支（见图 5-8c），当转换的实现导致几个序列同时激活时，这些序列称为并行序列。当步 3 是活动的，并且转换条件 e 为 1 状态，步 4 和步 6 同时变为活动步，同时步 3 变为不活动步。为了强调转换的同步实现，水平连线用双线表示。步 4 和步 6 被同时激活后，每个序列中活动步的进展将是独立的。在表示同步的水平双线之上，只允许有一个转换符号。

图 5-8　单序列、选择序列与并行序列

并行序列的结束称为合并（见图 5-8c），在表示同步的水平双线之下，只允许有一个转换符号。当直接连在双线上的所有前级步（步 5 和步 7）都处于活动状态，并且转换条件 i 为 1 状态时，才会发生步 5 和步 7 到步 10 的进展，即步 5 和步 7 同时变为不活动步，而步 10 变为活动步。

2．专用钻床的顺序功能图

某专用钻床用来加工圆盘状零件上均匀分布的 6 个孔，图 5-9 左边的示意图上面是侧视图，下面是工件的俯视图。顺序功能图包含了选择序列和并行序列。

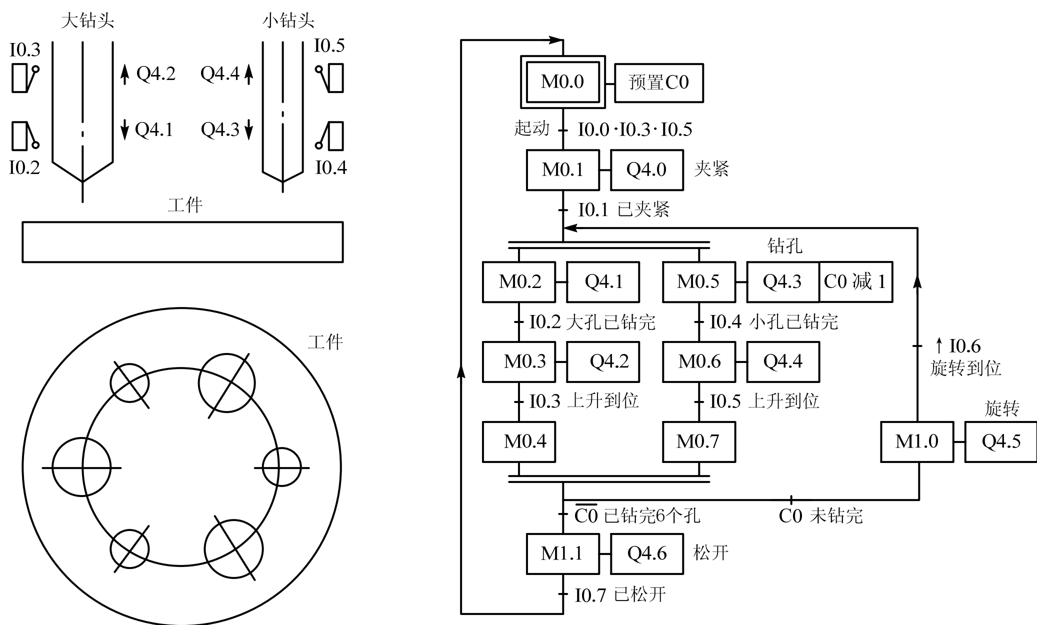


图 5-9　专用钻床控制系统的顺序功能图

在进入自动运行之前，两个钻头应在最上面，上限位开关 I0.3 和 I0.5 为 1 状态，系统处于初始步，减计数器 C0 的设定值 3 被送入计数器。操作人员放好工件后，按下起动按钮 I0.0，转换条件 I0.0·I0.3·I0.5 满足，由初始步转换到步 M0.1，Q4.0 变为 1 状态，工件被夹紧。夹紧后压力继电器 I0.1 为 1 状态，由步 M0.1 转换到步 M0.2 和 M0.5，Q4.1 和 Q4.3 使两只钻头同时开始向下钻孔，同时 C0 的当前值减 1。大钻头钻到由限位开关 I0.2 设定的深度时，进入步 M0.3，Q4.2 使大钻头上升，升到由限位开关 I0.3 设定的起始位置时停止上升，进入等待步 M0.4。小钻头钻到由限位开关 I0.4 设定的深度时，进入步 M0.6，Q4.4 使小钻头上升，升到由限位开关 I0.5 设定的起始位置时停止上升，进入等待步 M0.7。

C0 减 1 后当前值为 2（非 0），C0 的常开触点闭合，转换条件 C0 满足。两个钻头都上升到位后，将转换到步 M1.0。Q4.5 使工件旋转 120°，旋转到位时 I0.6 为 1 状态，又返回步 M0.2 和 M0.5，开始钻第二对孔。转换条件“↑I0.6”中的“↑”表示转换条件仅在 I0.6 的上升沿时有效。

3 对孔都钻完后，计数器的当前值变为 0，其常闭触点闭合，转换条件 $\overline{C0}$ 满足，进入步 M1.1，Q4.6 使工件放开。放开到位时，限位开关 I0.7 为 1 状态，系统返回初始步 M0.0。

因为要求两个钻头向下钻孔和钻头提升的过程同时进行，采用并行序列来描述上述的过程。由 M0.2～M0.4 和 M0.5～M0.7 组成的两个单序列分别用来描述大钻头和小钻头的工作过程。在步 M0.1 之后，有一个并行序列的分支。当 M0.1 为活动步，且转换条件 I0.1 得到满足（I0.1 为 1 状态），并行序列的两个单序列中的第 1 步（步 M0.2 和 M0.5）同时变为活动步。此后两个单序列内部各步的活动状态的转换是相互独立的，例如大孔或小孔钻完时的转换一般不是同步的。

两个单序列的最后 1 步（步 M0.4 和 M0.7）应同时变为不活动步。但是两个钻头一般不会同时上升到位，不可能同时结束运动，所以设置了等待步 M0.4 和 M0.7，它们用来同时结束两个并行序列。当两个钻头均上升到位，限位开关 I0.3 和 I0.5 分别为 1 状态，大、小钻头两个子系统分别进入两个等待步 M0.4 和 M0.7，并行序列将会立即结束。

在步 M0.4 和 M0.7 之后，有一个选择序列的分支。没有钻完 3 对孔时 C0 的常开触点闭合，转换条件 C0 满足，如果两个钻头都上升到位，将从步 M0.4 和 M0.7 转换到步 M1.0。如果已经钻完了 3 对孔，C0 的常闭触点闭合，转换条件 $\overline{C0}$ 满足，将从步 M0.4 和 M0.7 转换到步 M1.1。

在步 M0.1 之后，有一个选择序列的合并。当步 M0.1 为活动步，而且转换条件 I0.1 得到满足（I0.1 为 1 状态），将转换到步 M0.2 和 M0.5。当步 M1.0 为活动步，而且转换条件↑I0.6 得到满足，也会转换到步 M0.2 和 M0.5。

3. OB1 中的程序

用新建项目向导生成名为“钻床控制”的项目（见随书光盘中的同名例程），CPU 为 CPU 315-2DP。

图 5-10 是 OB1 中的程序，符号名为“自动开关”的 I2.0 为 1 状态时调用自动程序 FC1，为 0 状态时调用手动程序 FC2。在手动方式时，将各步对应的存储器位（M0.0～M1.7）复位，然后将初始步 M0.0 置位。上述操作主要是防止由自动方式切换到手动方式，然后又返回自动方式时，可能会出现同时有两个活动步的异常情况。



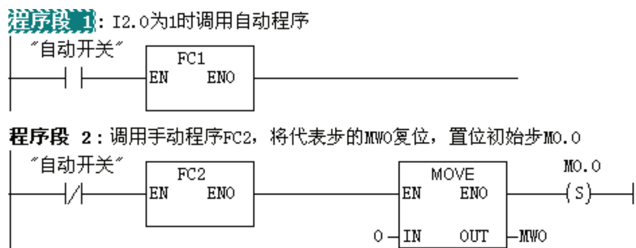


图 5-10　OB1 中的程序

执行 SIMATIC 管理器的菜单命令"插入"→"S7 块"→"组织块"，在出现的"属性–组织块"对话框中，生成组织块 OB100，设置"创建语言"为 LAD（梯形图）。用同样的方法生成自动程序 FC1 和手动程序 FC2。OB100 中的程序将除初始步 M0.0 之外的其他步清零（见图 5-11）。

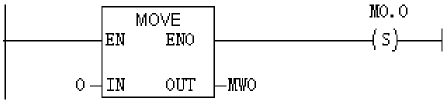


图 5-11　OB100 中的程序

程序中变量的符号见图 5-12 中的符号表。

	符号	地址	数据类型
1	起动按钮	I 0.0	BOOL
2	已夹紧	I 0.1	BOOL
3	大孔钻完	I 0.2	BOOL
4	大钻升到位	I 0.3	BOOL
5	小孔钻完	I 0.4	BOOL
6	小钻升到位	I 0.5	BOOL
7	旋转到位	I 0.6	BOOL
8	已松开	I 0.7	BOOL
9	大钻升按钮	I 1.0	BOOL
10	大钻降按钮	I 1.1	BOOL
11	小钻升按钮	I 1.2	BOOL
12	小钻降按钮	I 1.3	BOOL
13	正转按钮	I 1.4	BOOL
14	反转按钮	I 1.5	BOOL
15	夹紧按钮	I 1.6	BOOL
16	松开按钮	I 1.7	BOOL
17	自动开关	I 2.0	BOOL
18	夹紧阀	Q 4.0	BOOL
19	大钻头下	Q 4.1	BOOL
20	大钻头上	Q 4.2	BOOL
21	小钻头下	Q 4.3	BOOL
22	小钻头上	Q 4.4	BOOL
23	工件正转	Q 4.5	BOOL
24	松开阀	Q 4.6	BOOL
25	工件反转	Q 4.7	BOOL

图 5-12　符号表

4．手动程序

图 5-13 是 FC2 中的手动程序，为了节约篇幅，删除了各程序段的标题。在手动方式，用 8 个手动按钮分别独立操作大、小钻头的升降、工件的旋转和夹紧、松开。每对相反操作的输出点用对方的常闭触点实现互锁，用限位开关对钻头的升降限位。

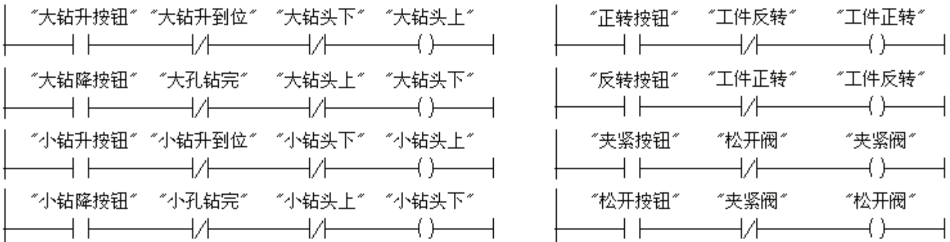


图 5-13　FC2 中的手动程序

5．自动控制程序

钻床控制的顺序功能图重画在图 5-14 中，图 5-15 是 FC1 中用置位复位指令编写的控

制代表步的存储器位的梯形图。

图 5-14 中分别由 M0.2～M0.4 和 M0.5～M0.7 组成的两个单序列是并行工作的，设计梯形图时应保证这两个序列同时开始工作和同时结束，即两个序列的第一步 M0.2 和 M0.5 应同时变为活动步，两个序列的最后一步 M0.4 和 M0.7 应同时变为不活动步。

并行序列的分支的处理是很简单的，在图 5-14 中，当步 M0.1 是活动步，并且转换条件 I0.1 为 1 状态时，步 M0.2 和 M0.5 同时变为活动步，两个序列开始同时工作。在图 5-15 的梯形图中，用 M0.1 和 I0.1 的常开触点组成的串联电路来控制对 M0.2 和 M0.5 的同时置位，以及对前级步 M0.1 的复位。

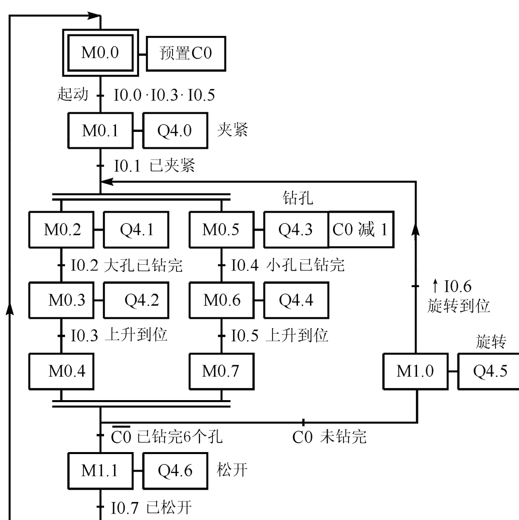


图 5-14　专用钻床控制系统的顺序功能图

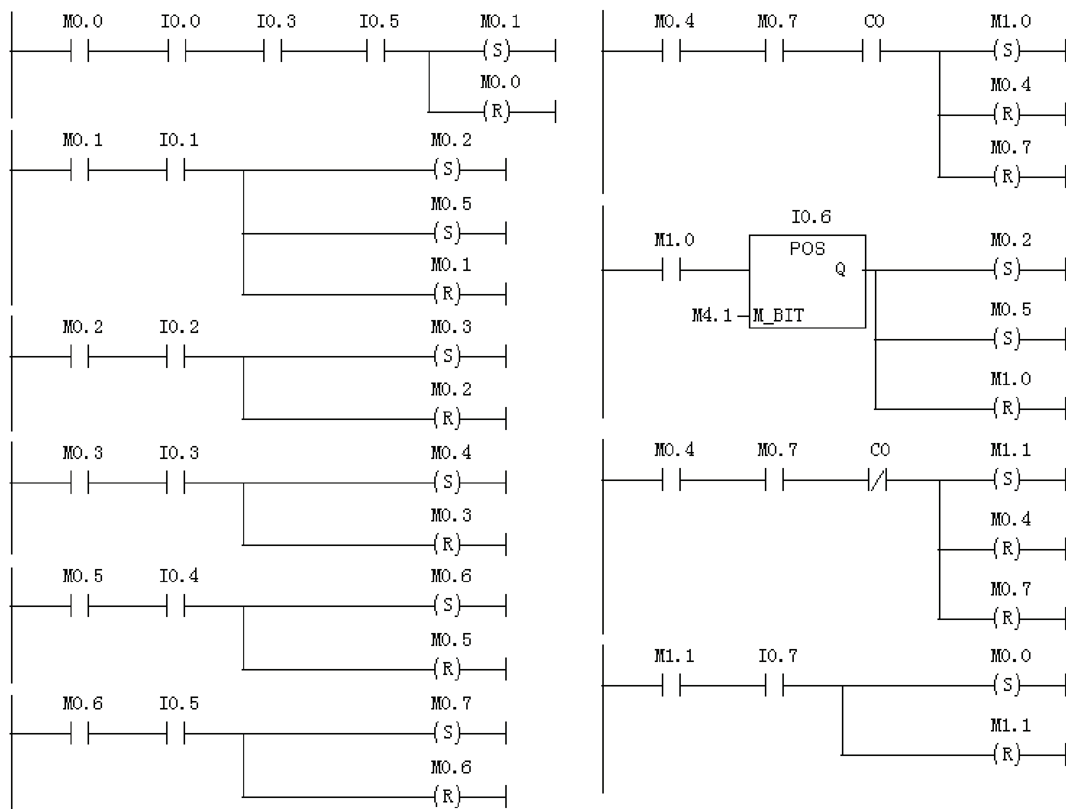


图 5-15　FC1 的顺序控制程序

另一种情况是当步 M1.0 为活动步，并且在转换条件 I0.6 的上升沿时，步 M0.2



和 M0.5 也应同时变为活动步，两个序列开始同时工作。在梯形图中，用 M1.0 和 I0.6 的上升沿检测指令组成的串联电路，来控制对 M0.2 和 M0.5 的同时置位，以及对前级步 M1.0 的复位。

图 5-14 的并行序列合并处的转换有两个前级步 M0.4 和 M0.7，当它们均为活动步并且转换条件满足时, 将实现并行序列的合并。未钻完 3 对孔时，减计数器 C0 的当前值非 0，其常开触点闭合，转换条件 C0 满足，将转换到步 M1.0。在梯形图中，用 M0.4、M0.7 和 C0 的常开触点组成的串联电路将 M1.0 置位，使后续步 M1.0 变为活动步；同时用 R 指令将 M0.4 和 M0.7 复位，使前级步 M0.4 和 M0.7 变为不活动步。

钻完 3 对孔时，C0 的当前值减至 0，其常闭触点闭合，转换条件$\overline{\text{C0}}$满足，将转换到步 M1.1。在梯形图中，用 M0.4、M0.7 的常开触点和 C0 的常闭触点组成的串联电路将 M1.1 置位，使后续步 M1.1 变为活动步；同时用 R 指令将 M0.4 和 M0.7 复位，使前级步 M0.4 和 M0.7 变为不活动步。

图 5-16 是自动程序 FC1 中用代表步的存储器位 M 控制各输出点 Q 和 C0 的输出电路。

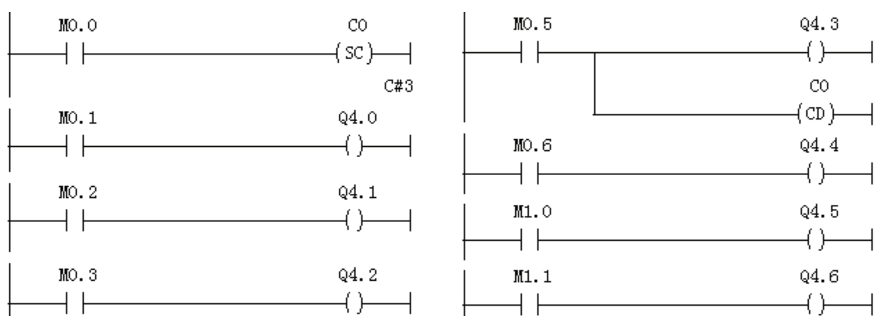


图 5-16　FC1 的输出电路

6．复杂的顺序程序的调试方法

调试复杂的顺序功能图时，应充分考虑各种可能的情况，对系统的各种工作方式、顺序功能图中的每一条支路、各种可能的进展路线，都应逐一检查，不能遗漏。特别要注意并行序列中各子序列的第 1 步（图 5-14 中的步 M0.2 和步 M0.5）是否同时变为活动步，最后一步（步 M0.4 和步 M0.7）是否同时变为不活动步。经过 3 次循环后，是否能进入步 M1.1，最后返回初始步。发现问题后应及时修改程序，直到每一条进展路线上步的活动状态的顺序变化和输出点的变化都符合顺序功能图的规定。

7．调试手动程序

打开 PLCSIM，生成与调试有关的视图对象（见图 5-17）。将各逻辑块下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。由于执行了 OB100 中的程序，初始步对应的 M0.0 为 1 状态，其余的步对应的存储器位为 0 状态。

令 I2.0 为 0 状态，CPU 调用手动程序 FC2，根据图 5-13 调试手动程序。手动程序采用点动控制，分别令各手动控制按钮 I1.0~I1.7 为 1 状态，观察对应的输出点是否为 1 状态。对大、小钻头作升、降控制时，观察对应的限位开关是否起作用。





图 5-17　PLCSIM

8．调试自动程序

令 I2.0 为 1 状态，CPU 调用自动程序 FC1。根据顺序功能图调试自动程序。进入自动程序时，仅初始步对应的 M0.0 为 1 状态。单击 PLCSIM 中 I0.3 和 I0.5 对应的小方框，模拟大、小钻头均在上限位置。单击两次 I0.0 对应的小方框，模拟按下和放开起动按钮。初始步下面的转换条件满足，M0.0 变为 0 状态，M0.1 和 Q4.0 变为 1 状态，说明转换到了夹紧步。令 I0.1 为 1 状态，转换到步 M0.2 和步 M0.5，C0 的当前值减 1 后变为 2。按照顺序功能图，依次令当前的活动步后面的转换条件为 1 状态，观察是否能转换到后续步。大小钻头均上升到位时，观察是否能转换到旋转步 M1.0，旋转到位时是否能返回步 M0.2 和步 M0.5。钻完 3 对孔后，观察是否能转换到步 M1.1，I0.7 为 1 状态时是否能返回初始步。

在调试时应注意在工件旋转期间，上限位开关 I0.3 和 I0.5 应为 1 状态，在钻孔期间，旋转到位开关 I0.6 应为 1 状态。

在任意的步为活动步时切换到手动方式（令 I2.0 为 0 状态），当前的活动步对应的存储器位（M）和输出点应变为 0 状态，初始步对应的 M0.0 应变为 1 状态。

9．顺序控制的仿真练习

3 条运输带顺序相连（见图 3-36），按下起动按钮 I0.2，1 号运输带开始运行，5s 后 2 号运输带自动起动，再过 5s 后 3 号运输带自动起动。停机的顺序与起动的顺序刚好相反，即按了停止按钮 I0.1 后，先停 3 号运输带，5s 后停 2 号运输带，再过 5s 停 1 号运输带。Q4.2～Q4.4 分别控制 1~3 号运输带。

在顺序起动 3 条运输带的过程中，操作人员如果发现异常情况，可以由起动改为停车。按下停止按钮 I0.3 后，将已经起动的运输带停车，仍采用后起动的运输带先停车的原则。图 5-18 是满足上述要求的顺序功能图。

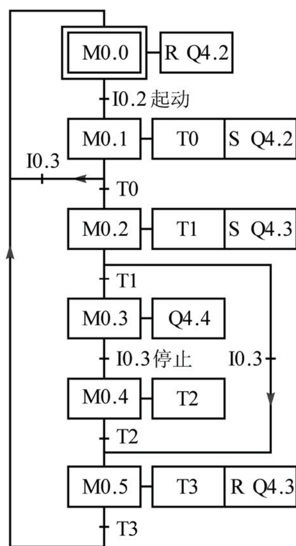


图 5-18　顺序功能图

根据顺序功能图，设计出梯形图程序，仔细检查后下载到 PLC。

调试步骤如下：



1）从初始步开始，按正常起动和停车的顺序调试程序。即从初始步 M0.0 开始，按步 M0.0、M0.1、M0.2、M0.3、M0.4 和 M0.5 的顺序调试程序，最后返回初始步。

2）从初始步开始，模拟调试在起动了一条运输带时停机的过程。即在第 2 步 M0.1 为活动步时，单击两次 I0.3 对应的小方框，模拟按下和放开停止按钮，观察是否能返回初始步。

3）从初始步开始，模拟调试在起动了两条运输带时停机的过程。即在第 3 步 M0.2 为活动步时，单击两次 I0.3 对应的小方框，模拟按下和放开停止按钮，观察是否能跳过步 M0.3 和步 M0.4，进入步 M0.5，经过 T3 设置的时间后，是否能返回初始步。

5.2　S7-GRAPH 顺序功能图语言的应用

5.2.1　实训三十一　使用 S7-GRAPH 的运输带顺序控制

S7 GRAPH 语言是 S7-300/400 用于顺序控制编程的顺序功能图语言，遵从 IEC 61131-3 标准中的顺序功能图语言“Sequential Function Chart”的规定。

1．安装 S7 GRAPH 语言

应在安装了 STEP 7 之后安装 S7-GRAPH。双击随书光盘的文件夹“\S7-GRAPH_V53\Disk1”中的文件“Setup.exe”，开始安装 S7-GRAPH。采用出现的“Setup Language”（安装语言）对话框（见图 5-19）中默认的语言 English。多次单击“Next”按钮，依次出现“Welcome”（欢迎）、Readme（自述文件）、Requirements（安装的要求）对话框。单击 Start Installation（启动安装）对话框中的“Next”按钮，开始安装软件。

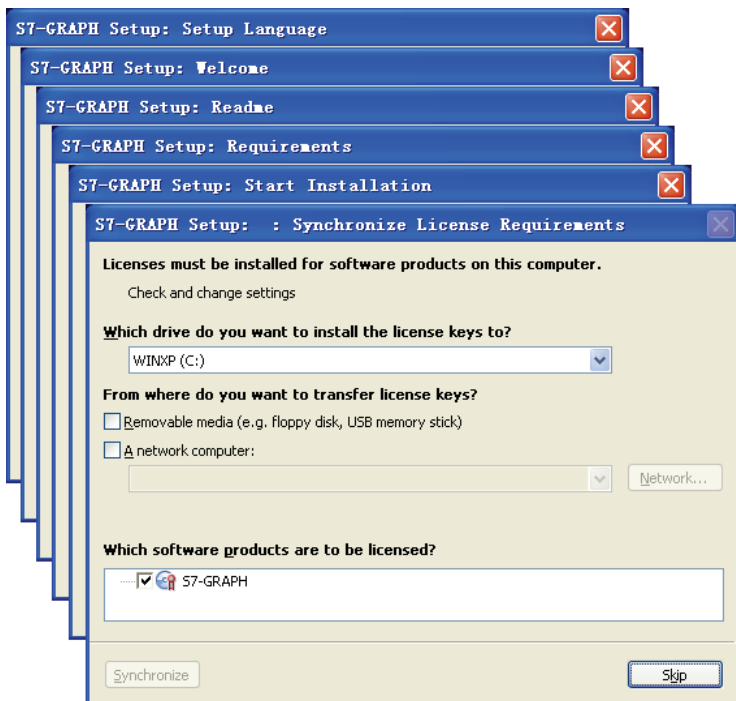


图 5-19　安装 S7-GRAPH 的对话框

安装过程中出现“Synchronize License Requirements”对话框（见图 5-19），要求安装许可证密钥。单击“Skip”按钮，跳过这一步骤。

最后出现“Setup is complete”（安装完成）对话框（见图 5-20），单击“Finish”按钮，安装结束。不用重新启动计算机，就可以使用 S7-GRAPH。

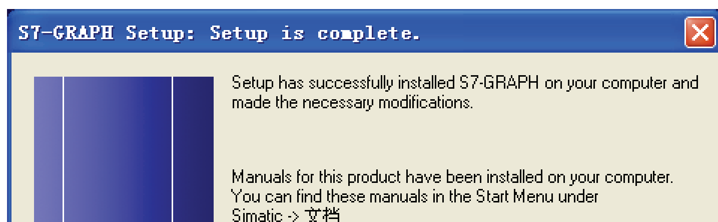


图 5-20　安装完成对话框

2. 创建使用 S7-GRAPH 的功能块

用新建项目向导生成名为“运输带 GR”的项目（见随书光盘中的同名例程），CPU 为 CPU 315-2DP。

执行 SIMATIC 管理器的菜单命令“插入”→“S7 块”→“功能块”，在出现的“属性 – 功能块”对话框中（见图 5-21），功能块默认的名称为 FB1，用下拉式列表设置“创建语言”为 GRAPH（即 S7-GRAPH）。

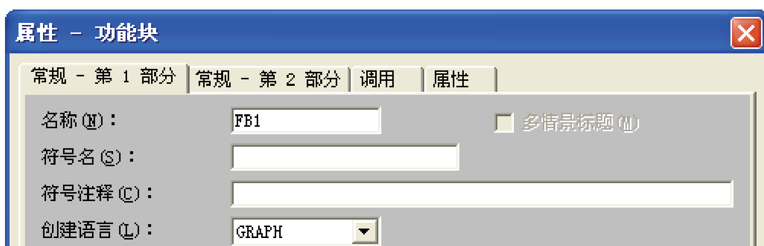


图 5-21　生成功能块

双击打开生成的 FB1，第一次打开 S7-GRAPH 编辑器时，出现图 5-22 所示的对话框，选中“S7-GRAPH”，“激活”按钮上的字符变为黑色，单击该按钮，激活期限为 14 天的试用许可证密钥。

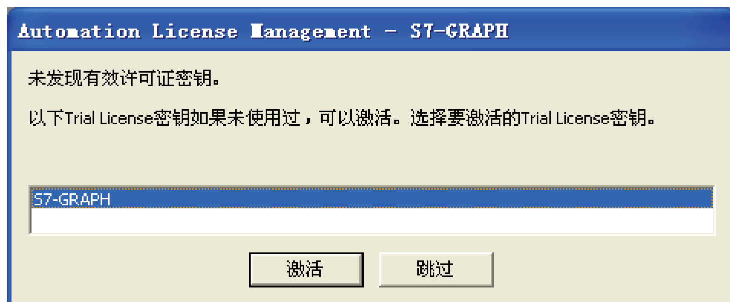


图 5-22　激活 S7-GRAPH 的试用许可证密钥



打开 FB1 后，右边的程序区有自动生成的步 S1 和转换 T1（见图 5-23）。

3．S7-GRAPH 编辑器

单击工具栏上的按钮（见图 5-23），可以打开或关闭块注释。最左边的顺序器工具栏可以拖到程序区的任意位置水平放置。单击按钮，可以关闭左边的浏览窗口和下面的详细窗口。

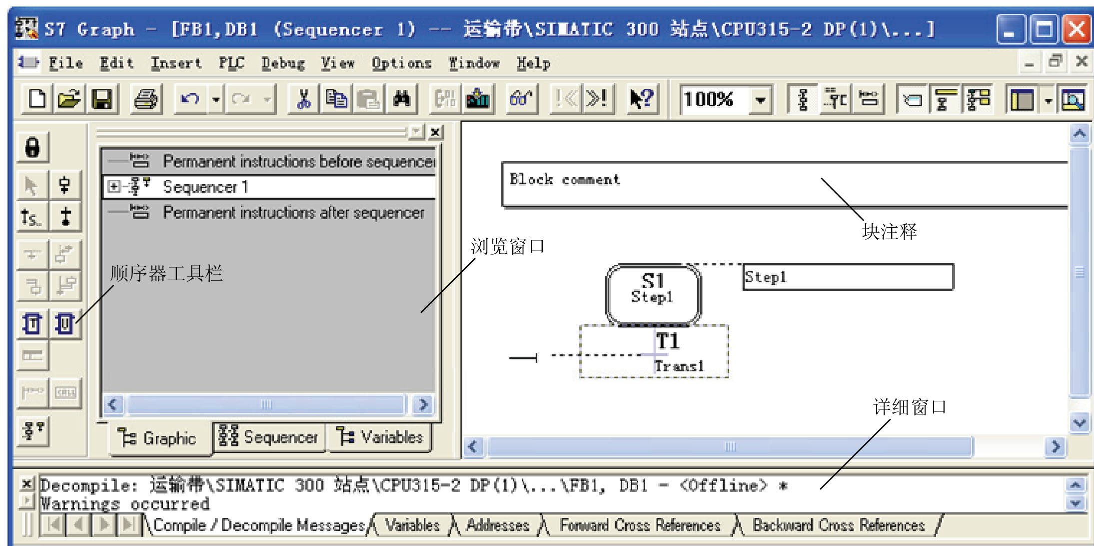


图 5-23　S7-GRAPH 的界面

浏览窗口中的“Graphics”（图形）选项卡的中间是顺序器（Sequencers），它的上面和下面是永久性指令（Permanent Instructions）。“Sequencers”选项卡用来浏览顺序器的总体结构，和选择显示哪一个顺序器。“Variables”（变量）选项卡中的变量是编程时可能用到的各种元素。可以在变量选项卡定义、编辑和修改变量。

4．生成步与转换

图 5-24 的中间的图是图 5-1 中运输带控制的顺序功能图，Q4.0 在连续的 3 步均为 1 状态。为了简化顺序功能图，在步 M0.1 用动作中的命令“S”将 Q4.0 置为 1，在初始步用动作中的命令“R”将 Q4.0 复位为 0（见图 5-24 右边的顺序功能图）。

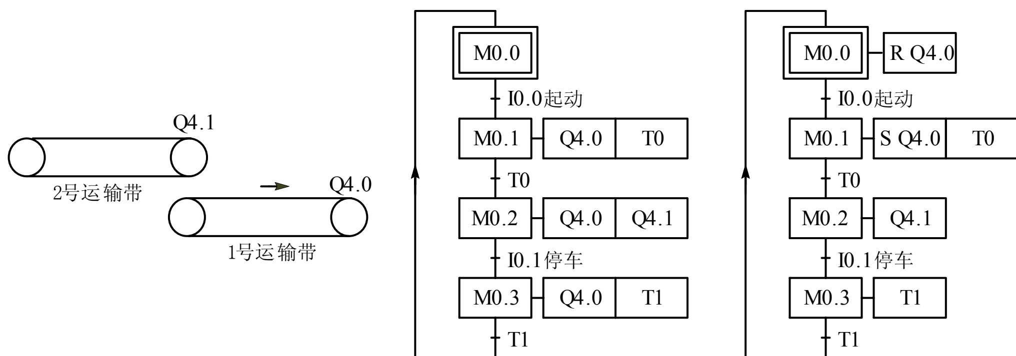


图 5-24　运输带控制的顺序功能图

单击按钮，隐藏动作和转换条件，隐藏后只显示步和转换。

选中图 5-25 的转换 T1，它变为浅紫色，周围出现虚线框。单击 3 次顺序器工具栏上的按钮，在 T1 的下面生成步 S2~S4 和转换 T2~T4（见图 5-25 a 中的顺序功能图），此时 T4 被自动选中。单击顺序器工具栏上的（Jump，跳转）按钮，在 T4 的下面出现一个箭头。在箭头旁的文本框中输入 1，表示将从转换 T4 跳转到初始步 S1。按计算机的回车键，在步 S1 上面的有向连线上，自动出现一个水平的箭头（见图 5-25b），它的右边标有转换 T4，相当于生成了一条起于 T4，止于步 S1 的有向连线。至此步 S1～S4 形成了一个闭环。

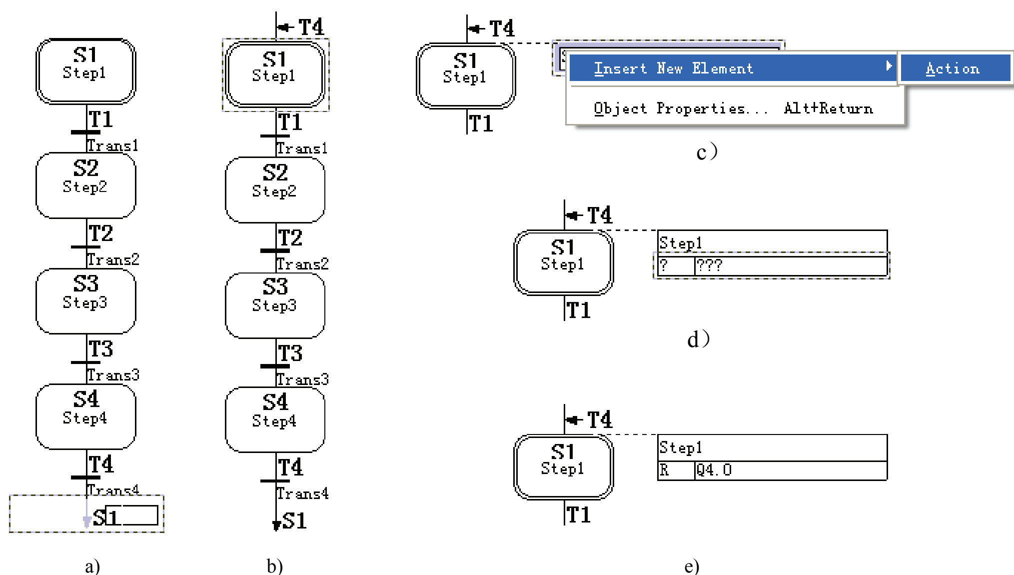


图 5-25　生成跳步与动作

代表步的方框内有步的编号（例如 S2）和名称（例如 Step2），单击选中后，可以修改它们，不能用汉字作步和转换的名称。用同样的方法，可以修改转换的编号（例如 T2）和名称（例如 Trans2）。单击步的编号和名称之外的其他部分，表示步的方框整体变色，称为选中了该步。

5．生成动作

单击按钮，显示被隐藏的动作和转换条件。

用鼠标右键单击初始步 S1 右边的动作框，执行出现的快捷菜单中的命令“Insert New Element”（插入新元件）→“Action”（动作，见图 5-25c），插入一个空的动作行。

一个动作行由命令和地址组成，它左边的方框用来写入命令，下面是一些常用的命令：

1）命令 S：当步为活动步时，使输出置位为 1 状态并保持。

2）命令 R：当步为活动步时，使输出复位为 0 状态并保持。

3）命令 N：当步为活动步时，输出为 1；该步变为不活动步时，输出被复位为 0。

4）命令 D：使某一动作的执行延时，延时时间在该命令右下方的方框中设置，T#5S 表示延时 5s。延时时间到时，如果步仍然保持为活动步，则使该动作输出为 1；如果该步已变



为不活动步，该动作的输出为 0。

5）命令 CALL：当该步为活动步时，调用命令中指定的逻辑块。

单击图 5-25d 的动作框中的“？”，输入动作的命令“R”。单击动作框中的“???”，输入动作的地址“Q4.0”（见图 5-25e），在初始步将 Q4.0 复位。用同样的方法，在步 S2 生成一个使 Q4.0 置位的动作（见图 5-26）。

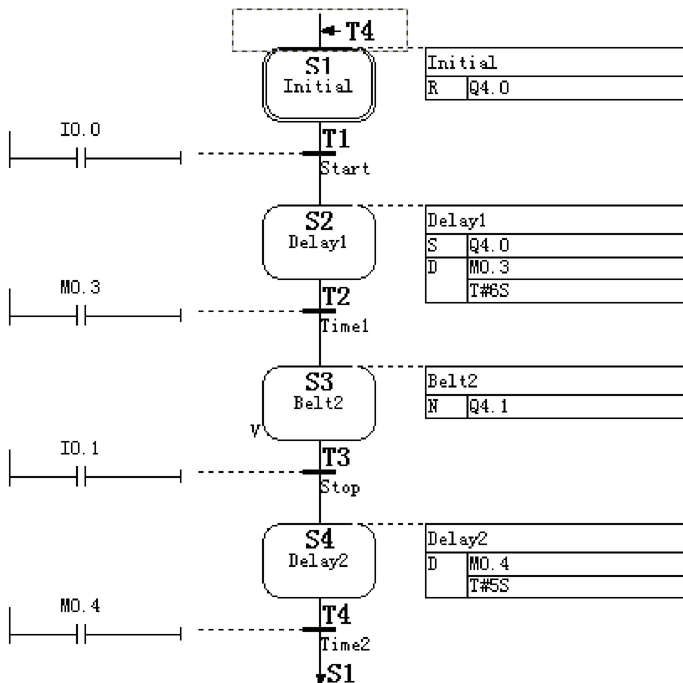


图 5-26　顺序功能图

在步 S2 的动作框中输入命令“D”后，命令框的右边自动出现两行，在上面一行输入地址 M0.3，下面一行输入“T#6S”（延时时间为 6s）。延时时间到时，M0.3 变为 1 状态，步 S2 之后的转换条件满足。用上述的方法，生成其余各步的动作。

执行菜单命令“View”→“Display with”→“Symbols”，可以在符号显示和绝对地址显示之间切换。

6．S7-GRAPH 编辑器的参数设置

执行菜单命令“Options”（选项）→“Application Settings”（应用设置），单击打开的对话框的“General”选项卡中的“Comments”（注释）复选框（见图 5-27），去掉其中的☑，生成新的 S7-GRAPH 功能块时将会没有注释。用单选框选中“Conditions in new block”区的“LAD”，新的块的转换条件默认的语言为梯形图。

S7-GRAPH 默认的转换条件的字符太小，用“Editor”（编辑器）选项卡中的 Font（字体）区的“Object type”（对象类型）下拉式列表选中“LAD/FDB”（见图 5-27 左下角的图），单击“Select…”（选择）按钮，打开“字体”对话框（见图 5-27 右边的图），设置字体的大小为 14，单击“确定”按钮和“OK”按钮确认。

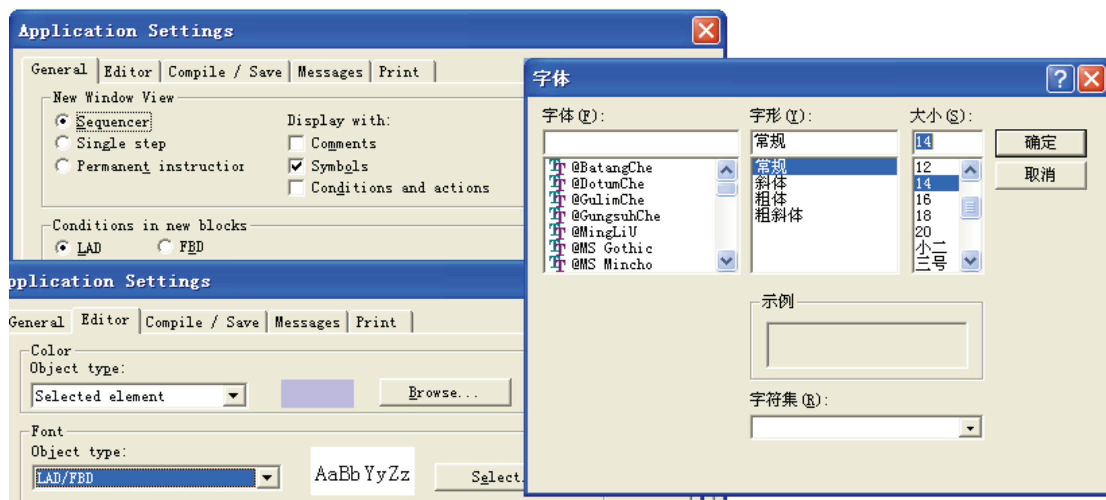


图 5-27　S7-GRAPH 编辑器的参数设置

7. 生成转换条件

转换条件可以用梯形图或功能块图来表示。转换条件工具栏在编辑器最左边，顺序器工具栏的下面，图 5-28a 是使用梯形图（LAD）语言的转换条件工具栏。初次打开 S7-GRAPH 编辑器时，默认的是使用功能块图（FBD）的转换条件工具栏。可以用“View”菜单中的命令切换这两种语言。选中转换 T1 对应的转换条件（见图 5-28b），单击工具栏上的按钮，T1 的转换条件出现一个常开触点（见图 5-28c）。单击触点上面红色的??.?，输入地址 I0.0（见图 5-28d）。用同样的方法生成其他转换条件。

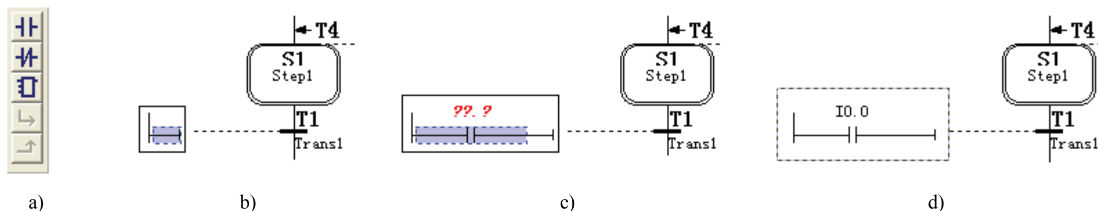


图 5-28　生成转换条件

转换条件可以是多个触点和比较器（对应于工具栏上的按钮）的串并联电路。比较器相当于一个触点。

8. 对监控功能编程

双击步 S3，切换到单步视图（见图 5-29），选中 Supervision（监控）线圈，单击图 5-28a 的工具栏上的比较器按钮，在比较器左边中间的引脚输入“S3.T”（步 S3 为活动步的时间），在比较器左边下面的引脚输入时间预置值“T#2H”，设置的监视时间为 2h。如果该步的执行时间超过 2h，该步被认为出错，监控时出错的步用红色显示。选中比较器中间的比较符号“>”后，可以修改它。

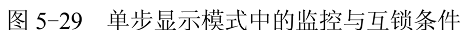


图 5-29 单步显示模式中的监控与互锁条件

9．设置 S7-GRAPH 功能块的参数集

执行菜单命令“Options”（选项）→“Block settings”（块设置），在打开的对话框的“FB Parameters”（FB 参数）区（见图 5-30），用单选框选中“Minimum”（最小参数集），此时 FB1 只有一个参数。单击“OK”按钮确认。

单击工具栏上的按钮，保存和编译 FB1 中的程序。如果程序有错误，下面的详细窗口将给出错误提示和警告，改正错误后才能保存。

10．调用 S7-GRAPH 功能块

双击打开 OB1，设置编程语言为梯形图。将指令列表的“FB 块”文件夹中的 FB1 拖放到程序段 1 的“电源线”上（见图 5-31）。

图 5-30 设置块的参数

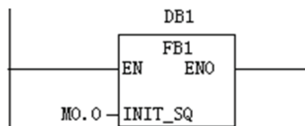


图 5-31 OB1 的程序

在 FB1 方框的上面输入它的背景数据块的编号 DB1，按回车键后出现的对话框询问“实例数据块 DB1 不存在，是否要生成它？”。单击“是”按钮确认。FB1 的形参 INIT_SQ 为 1 时，顺序器被初始化，仅初始步为活动步。

11．仿真实验

打开 PLCSIM，创建 IB0 和 MB0 的视图对象。将所有的块下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。打开 FB1，单击工具栏上的按钮，起动程序状态监控功能（见图 5-32）。

刚开始监控时只有初始步 S1 为绿色，表示它为活动步。该步的动作框上面的两个监控定时器开始定时。它们用来记录当前步被激活的时间，其中定时器 U 不包括有干扰的时间。

单击两次 PLCSIM 中 I0.0 对应的小方框，模拟按下和放开起动按钮。可以看到步 S1 变为白色，步 S2 变为绿色，表示由步 S1 转换到了步 S2。

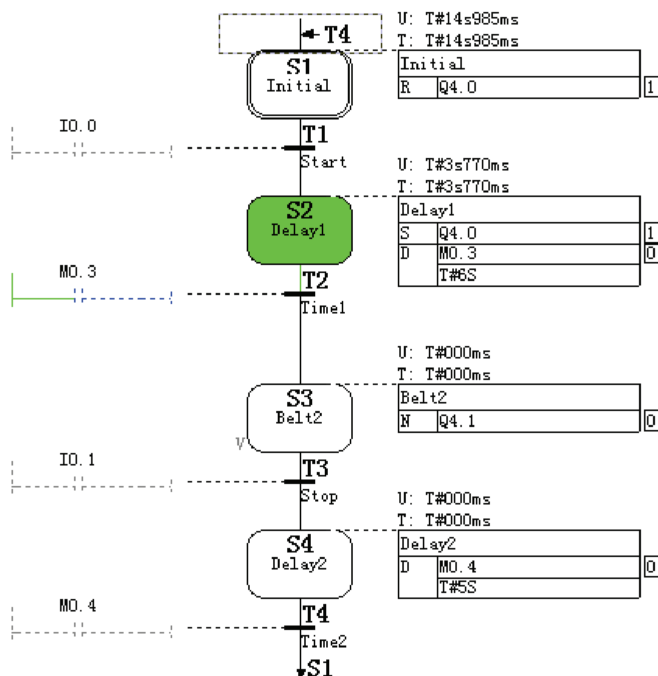


图 5-32　顺序功能图的程序状态

步 S2 的动作方框上面的监控定时器的当前时间值达到预置值 6s 时，M0.3 变为 1 状态，步 S2 下面的转换条件满足，将自动转换到步 S3。单击两次 I0.1 对应的小方框，模拟停止按钮的操作，将会观察到由步 S3 转换到步 S4，延时 5s 后自动返回初始步。

各个动作右边的小方框显示该动作的 0、1 状态。只显示活动步后面的转换条件中的能流。

单击两次 PLCSIM 中 M0.0 对应的小方框，给 OB1 中 FB1 的输入参数“INIT_SQ”提供一个脉冲。在脉冲的上升沿，顺序控制器被初始化，初始步 S1 变为活动步，其余各步为非活动步。

12．S7-GRAPH 的仿真练习

生成一个新的项目，在项目中生成使用 S7-GRAPH 语言的功能块 FB1，在 FB1 中生成图 5-3 中的冲床机械手控制系统的程序，在 OB1 中调用 FB1。将程序下载到仿真 PLC，调试程序，直到满足要求。

5.2.2　实训三十二　使用 S7-GRAPH 的剪板机顺序控制

1．剪板机的控制要求

图 5-33 是某剪板机的示意图，开始时压钳和剪刀在上限位置，限位开关 I0.0 和 I0.1 为 1 状态。系统的顺序功能图如图 5-34 所示。图中有选择序列、并行序列的分支与合并。

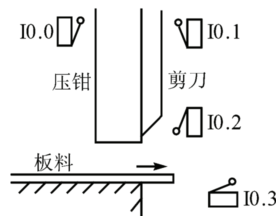


图 5-33　剪板机示意图

系统用连续方式开关 I0.7 来切换两种工作方式。I0.7 为 1 状态时，按下起动按钮 I0.5，连续标志 M0.1 变为 1 状态。首先板料右行至限位开关 I0.3 动作（见图 5-33），然后压钳下行，压紧板料后，压力继电器 I0.4 为 1 状态，压钳保持压紧，剪刀开始下



行。剪断板料后，I0.2 变为 1 状态，压钳和剪刀同时上行，它们碰到限位开关 I0.0 和 I0.1 后，分别进入等待步 S6 和 S8，停止上行。都进入等待步后，返回步 S2，又开始下一周期的工作。

按下停止按钮后，并不马上停止工作，要等到完成当前周期的全部任务，压钳和剪刀都上升到位后，才返回初始步。

在单周期方式，连续方式开关 I0.7 为 0 状态。按下起动按钮，只剪一块板料。完成当前周期的任务后，返回初始步。

图 5-34 中步 S4 之后有一个并行序列的分支，当步 S4 是活动步，并且转换条件 I0.2 满足，步 S5 与步 S7 同时变为活动步。与此同时，步 S4 变为不活动步。

步 S6 和 S8 是等待步，它们用来同时结束并行序列的两个子序列。步 S6 与步 S8 之后有一个并行序列的合并，只要步 S6 和 S8 都是活动步，就会根据连续标志 M0.1 的状态，分别转换到步 S1 或步 S2。

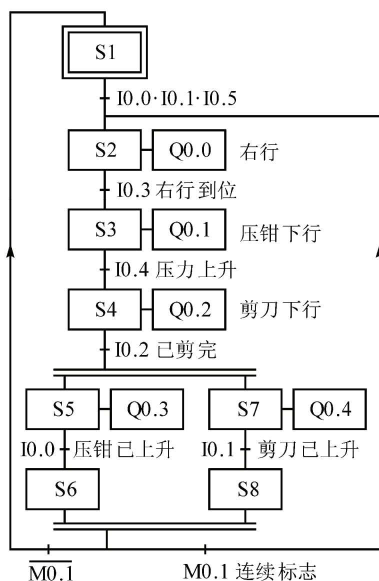


图 5-34　剪板机控制的顺序功能图

2. 创建使用 S7-GRAPH 的功能块

用新建项目向导生成名为“剪板机 GR”的项目（见随书光盘中的同名例程），CPU 为 CPU 315-2DP。

执行 SIMATIC 管理器的菜单命令“插入”→“S7 块”→“功能块”，在出现的“属性 – 功能块”对话框中（见图 5-21），功能块默认的名称为 FB1，用下拉式列表设置“创建语言”为 GRAPH。

3. 生成选择序列

双击打开 FB1，单击 S7-GRAPH 编辑器工具栏上的按钮，隐藏动作和转换条件，隐藏后只显示步和转换。选中自动生成的转换 T1（见图 5-23），它变为浅紫色，周围出现虚线框。单击 5 次顺序器工具栏上的按钮，在 T1 的下面生成步 S2~S6 和转换 T2~T6（见图 5-35）。选中步 S6，单击顺序器工具栏上的按钮，生成一条选择序列的分支和转换 T7。

选中转换 T6，单击顺序器工具栏上的跳转按钮，在出现的文本框中输入 1，从转换 T6 跳转到初始步 S1。按计算机的回车键，在 T6 的下面出现一个标有 S1 的箭头。与此同时，在步 S1 上面的有向连线上，自动出现一个水平的箭头，它的右边标有转换 T6，相当于生成了一条起于 T6，止于步 S1 的有向连线。

选中转换 T7，单击顺序器工具栏上的跳转按钮，在出现的文本框中输入 2，从转换 T7 跳转到步 S2。在步 S2 之前，有一个选择系列的合并。

4. 生成并行序列

选中 T4，单击顺序器工具栏上的按钮，生成并行序列的分支和步 S7。单击按钮，生成转换 T8 和步 S8。

此时自动选中了步 S8，单击顺序器工具栏上的按钮，出现从步 S8 下边沿中点开始的水平双线、按钮上的图形，和表示禁止放置的符号。将水平双线拖到步 S6 的下边沿，

禁止放置的符号消失。单击鼠标左键，S6 和 S8 被水平双线连接到一起。

5．生成动作与转换条件

单击 S7-GRAPH 编辑器工具栏上的按钮，显示各步的动作框和转换条件。用鼠标右键单击步 S2 右边的动作框，执行出现的快捷菜单中的命令“Insert New Element”（插入新元件）→“Action”（动作），插入一个空的动作行。输入该动作的命令“N”和地址 Q0.0（见图 5-36）。用同样的方法生成其余各步的动作。

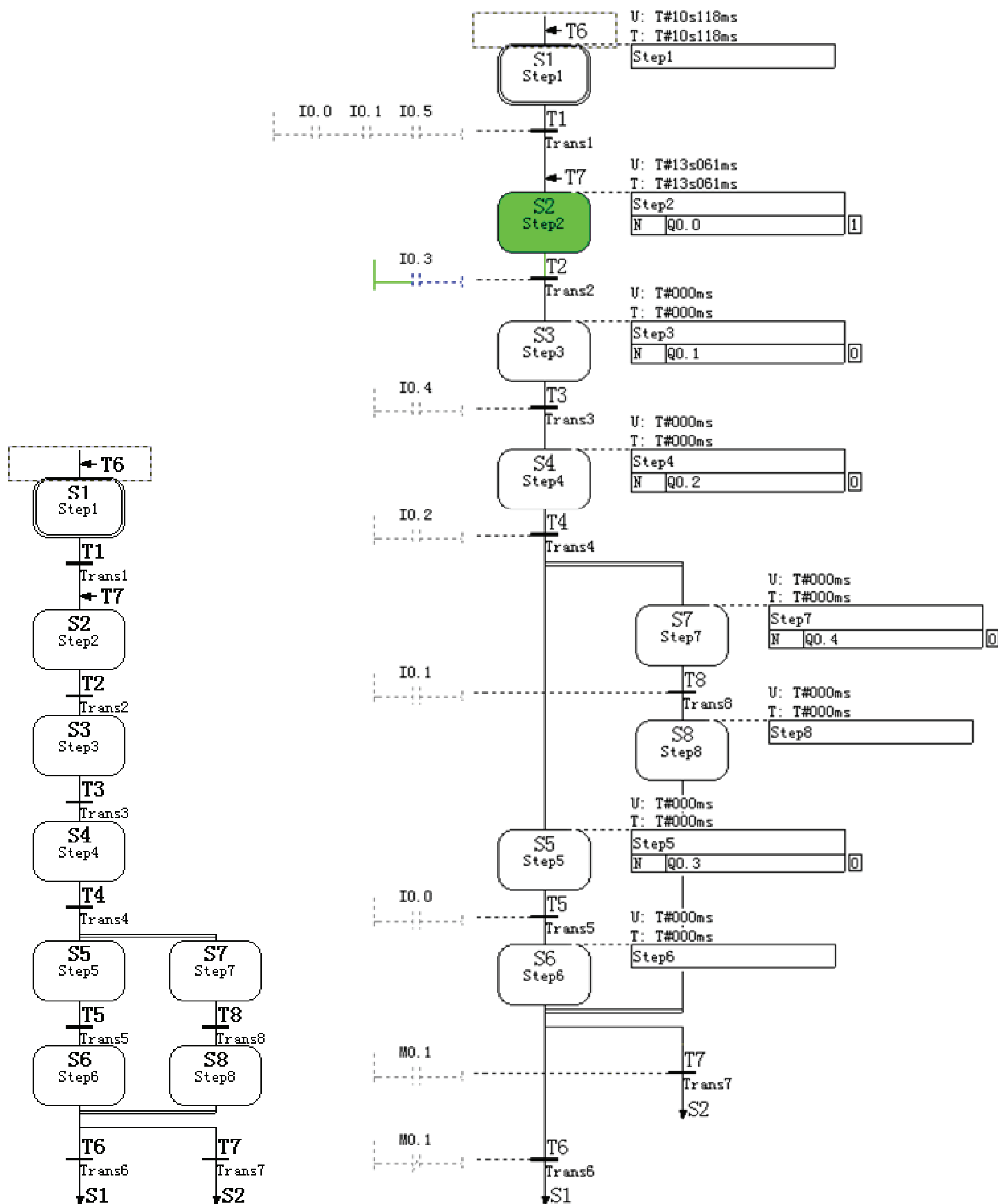


图 5-35　顺序功能图　　　　图 5-36　顺序功能图的程序状态



选中 T1 对应的转换条件，单击 3 次顺序器工具栏上的按钮，生成 3 个串联的常开触点。单击触点上面红色的??.?，输入各触点的地址。用同样的方法生成其他转换条件。

6．生成永久性指令

执行菜单命令“View”（视图）→“Overviews”（浏览），打开左边的浏览窗口（见图 5-23）。打开“Graphic”选项卡，单击其中的“Permanent instructions before sequencer”，打开顺序器之前的永久性指令视图（见图 5-38）。

执行菜单命令“Insert”（插入）→“Permanent instructions”（永久性指令）→“Conditions”（条件），出现一个线圈（见图 5-37a），选中它之后，单击顺序器工具栏上的按钮，生成两个常开触点。选中左边的垂直线，单击工具栏上的按钮，生成一条支路，单击按钮，支路上出现一个常开触点（见图 5-37b）。单击按钮，将 3 个触点并联在一起（见图 5-37c）。单击工具栏上的按钮，生成一个串联的常闭触点。

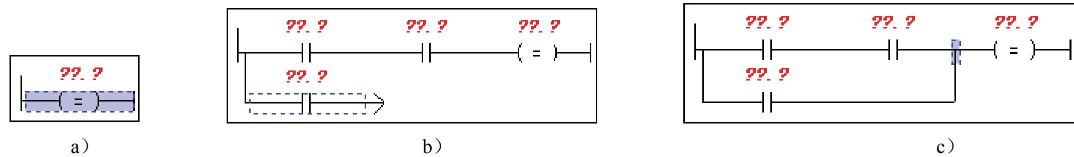


图 5-37　生成永久性指令

输入各触点和线圈的地址后的永久性指令如图 5-38 所示。在连续方式按下起动按钮，连续标志变为 1 状态。在单周期方式按下起动按钮，连续标志为 0 状态。

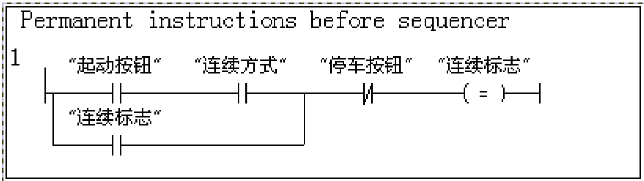


图 5-38　永久性指令

图 5-39 是项目的符号表。

符号编辑器 - [S7 程序(1) (符号) -- 剪板机GR\SIM...

符号表(S)　编辑(E)　插入(I)　视图(V)　选项(O)　窗口(W)　帮助(H)

	符号	地址		数据类型	注释
2	压钳上限	I	0.0	BOOL	
3	剪刀上限	I	0.1	BOOL	
4	剪刀下限	I	0.2	BOOL	
5	右限位	I	0.3	BOOL	
6	压力上升	I	0.4	BOOL	
7	起动按钮	I	0.5	BOOL	
8	停车按钮	I	0.6	BOOL	
9	连续方式	I	0.7	BOOL	
10	初始脉冲	M	0.0	BOOL	
11	连续标志	M	0.1	BOOL	

图 5-39　符号表

7．调用 S7-GRAPH 功能块

执行菜单命令“Options”（选项）→“Block Settings”（块设置），在打开的对话框的“FB Parameters”（FB 参数）区（见图 5-30），用单选框选中“Minimum”（最小参数集），单击“OK”按钮确认。单击工具栏上的按钮，保存和编译 FB1 中的程序。

打开 OB1，设置编程语言为梯形图。将指令列表的“FB 块”文件夹中的 FB1 拖放到程序段 1 的“电源线”上（见图 5-31）。

在 FB1 方框的上面输入它的背景数据块的编号 DB1，按回车键后出现是否要生成 DB1 的对话框，单击“是”按钮确认。

8．仿真实验

打开 PLCSIM，创建 IB0 和 MB0 的视图对象。将所有的块下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。打开 FB1，单击工具栏上的按钮，起动程序状态监控功能（见图 5-36）。可以看到仅初始步为活动步（用绿色表示）。

令“连续方式”开关 I0.7 为 0 状态，系统工作方式为单周期。单击视图对象 IB0 中的小方框，令 I0.0 和 I0.1 为 1 状态，模拟压钳和剪刀均在最上面。单击两次 I0.5 对应的小方框，模拟按下和放开起动按钮，连续标志 M0.1 仍为 0 状态。可以看到步 S1 变为白色，步 S2 变为绿色，表示由步 S1 转换到了步 S2，板料右行。单击两次 I0.3 对应的小方框，模拟右行到位，从步 S2 转换到步 S3，压钳下行。令压钳上限位开关 I0.0 为 0 状态，压力开关 I0.4 为 1 状态，从步 S3 转换到步 S4，剪刀下行。令剪刀上限位开关 I0.1 为 0 状态，剪刀下限位开关 I0.2 为 1 状态，从步 S4 转换到步 S5 和步 S7，压钳和剪刀同时上行。分别令上限位开关 I0.0 和 I0.1 为 1 状态，等待步 S6 和 S8 变为活动步。因为连续标志为 0 状态，转换条件 $\overline{\text{M0.1}}$ 满足，将返回初始步，结束一个周期的工作。

令“连续方式”开关 I0.7 为 1 状态，系统工作方式为连续。令 I0.0 和 I0.1 为 1 状态，按下起动按钮，板料开始右行。由于永久性指令的作用，连续标志 M0.1 为 1 状态。

用上述的方法，模拟剪完一块板料的过程，压钳和剪刀均上升到位后，观察是否从步 S6 和步 S8 返回到 S2，开始下一个周期的操作。

按下停车按钮 I0.6，连续标志 M0.1 变为 0 状态。观察在完成最后一个周期的工作后，是否返回到初始步。

9．S7-GRAPH 的仿真练习

图 5-18 中的 3 条运输带控制的顺序功能图重画在图 5-40。生成一个新的项目，在项目中生成使用 S7-GRAPH 语言的功能块 FB1，在 FB1 中生成满足图 5-40 要求的程序，在 OB1 中调用 FB1。将程序下载到仿真 PLC，调试程序，直到满足要求。

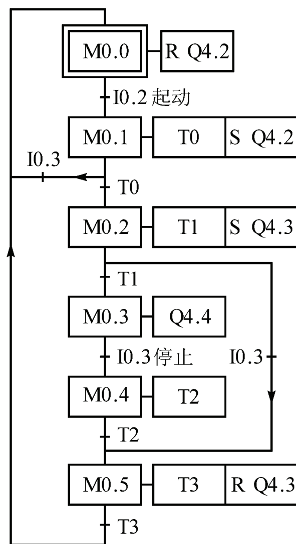


图 5-40　顺序功能图

5.3　实训三十三　生成与显示参考数据

STEP 7 为用户提供各种参考数据，参考数据对阅读和分析大型复杂的用户程序是非常



有用的。

1．显示参考数据

打开随书光盘中的项目“钻床控制”，用右键单击 SIMATIC 管理器左边窗口的“块”，执行出现的快捷菜单中的命令“参考数据”→“显示”。如果出现图 5-41 中的对话框，用单选框选中“更新”，单击“是”按钮，更新参考数据。更新完成后，出现图 5-42 左边的“自定义”对话框，在对话框中用单选框选择“交叉参考”，单击“确定”按钮，打开“参考”窗口，显示选中的交叉参考表（见图 5-42 中的右图）。

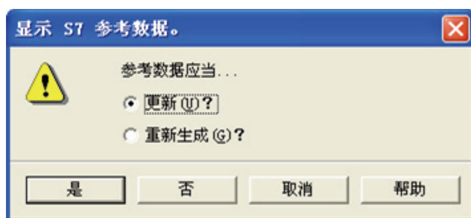


图 5-41　显示参考数据对话框

打开参考数据显示窗口后，可以用工具栏中对应的按钮显示其他参考数据。

执行参考数据显示窗口的菜单命令“窗口”→“新建窗口”，出现图 5-42 中左边的对话框，选择新窗口要显示的参考数据。可以同时打开多个参考数据窗口。

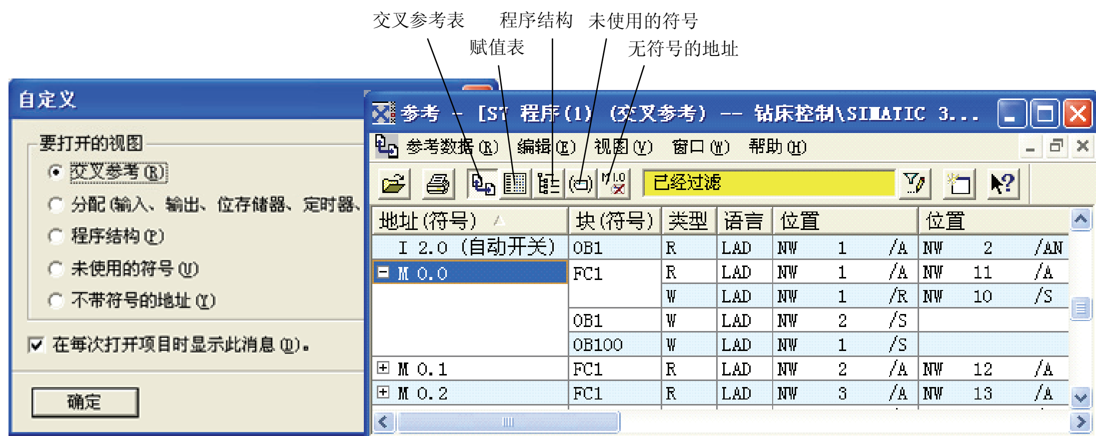


图 5-42　参考数据

2．交叉参考表

交叉参考表给出了 S7 用户程序使用的地址的概况，显示 I、Q、M、T、C、FB、FC、SFB、SFC、PI/PQ 和 DB 的绝对地址、符号地址，以及使用的情况。“类型”列的“R”和“W”分别表示读和写。“块”列是变量所在的程序块，“位置”列给出了变量在程序块中的位置和指令，例如“NW　1　/A”是程序段 1 中的“A”（与）指令。可以用鼠标左右拖动表格最上面灰色的表头中各列的分界点，来调节表格各列的宽度。

单击地址列左边的⊞，可以查看该地址被多次使用的情况。单击地址列的⊟，可以将同一地址有关的各行缩为一行。

执行菜单命令“编辑”→“查找”，可以搜索到指定的地址或符号。

3．交叉参考表的参数设置

执行菜单命令“视图”→“过滤”，将出现图 5-43 中的“过滤参考数据”对话框，可

以设置只显示部分地址。打开“交叉参考”选项卡，选中复选框“输入”，在它右边的输入框内输入“10-20; 24;”，表示输入的显示范围为 IB10～IB20 和 IB24。

进行下列的操作，观察操作对交叉参考表显示内容的影响：

1）在“位存储器”文本框输入 1，单击“确定”按钮确认。

2）在“位存储器”文本框输入 0-1，单击“确定”按钮确认。

3）单击“位存储器”右边的复选框，去掉其中的“√”，单击“确定”按钮确认。

4）单击“帮助”按钮，阅读帮助信息，修改图 5-43 中其他的参数，观察操作的效果。

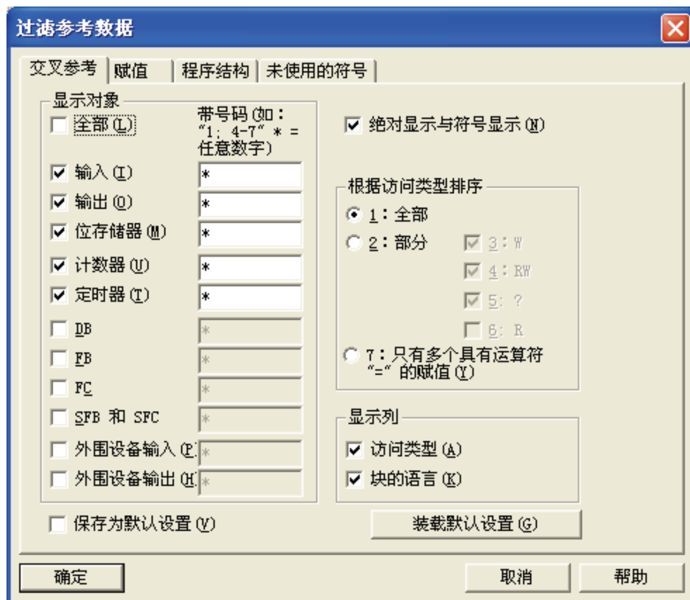
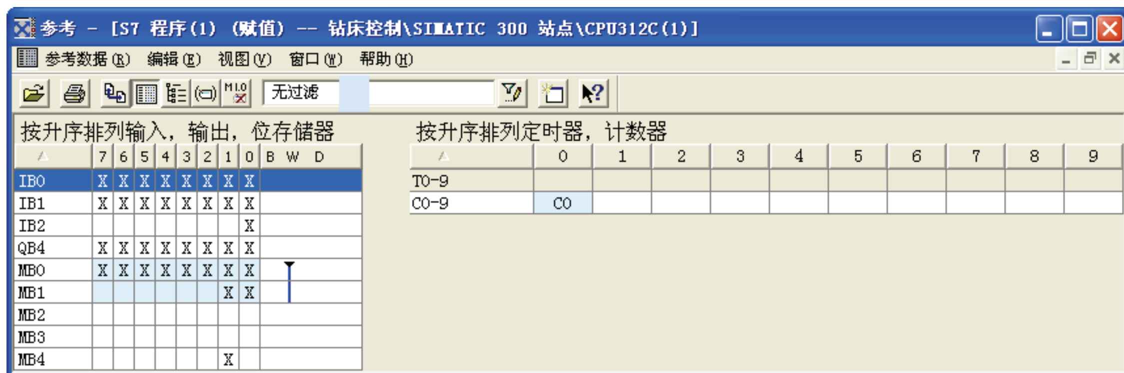


图 5-43　交叉参考表的参数设置

4．赋值表

赋值表（见图 5-44）显示已被用户程序使用的地址。



	7	6	5	4	3	2	1	0	B W D
IB0	X	X	X	X	X	X	X	X	
IB1	X	X	X	X	X	X	X	X	
IB2								X	
QB4	X	X	X	X	X	X	X	X	
MB0	X	X	X	X	X	X	X	X	
MB1							X	X	
MB2									
MB3									
MB4							X		

	0	1	2	3	4	5	6	7	8	9
T0-9										
C0-9	C0									

图 5-44　赋值表

赋值表的左边显示输入（I）、输出（Q）和位存储器（M）区哪些字节、哪些位被使



用，一个字节占一行，标有“X”的方格表示该位被访问。“B W D”列分别用来表示按什么存储单位（字节、字或双字）访问，例如图 5-44 中的 MB0 和 MB1 以字为单位访问。以字节、字或双字为单位访问的行用浅蓝色背景来表示。

赋值表的右边显示用户程序使用的定时器（T）和计数器（C），该项目只使用了计数器 C0。

5. 程序结构

程序结构显示用户程序中块的分层调用结构，通过它可以对程序所用的块、它们的从属关系以及它们对局部数据的需求有一个概括的了解（见图 5-45）。

打开随书光盘中的项目“S7_MPI”，用右键单击 SIMATIC 管理器左边窗口的“块”，执行出现的快捷菜单中的命令“参考数据”→“显示”。出现图 5-42 左边的“自定义”对话框时，用单选框选中“程序结构”，单击“确定”按钮，打开“参考”视图，显示该项目的程序结构（见图 5-45）。



图 5-45　程序结构窗口

“块（符号），实例数据块（符号）”列显示逻辑块、功能块的背景数据块，和逻辑块使用的共享数据块，以及它们的符号。

“语言”列是调用的块的编程语言。“局部数据（在路径中）”列显示调用结构中需要的最大的局部数据字节数，包括每个 OB 需要的最大局部数据和每个路径需要的局部数据。

SFB12 的“位置”列的“NW　1　Sta　1”表示它被程序段 1 的第一条指令调用。没有被调用的块在程序结构的底部显示，并且用黑叉标记。

在程序结构窗口作下列操作：

1）双击 OB1 所在的行，观察是否能打开 OB1。在 OB1 中观察调用 SFB12 和 SFB13 的程序段和语句的编号与程序结构中给出的是否相同。

2）双击程序结构列表中的 DB1，观察是否能打开它。

3）单击图 5-45 中 OB35 和 OB100 左边的⊞，了解它们调用块的情况。打开 OB35 和 OB100，观察程序结构给出的信息是否正确。

6. 其他参考数据

单击参考数据窗口工具栏上的“未使用的符号”按钮，可以显示在符号表中已经定

义，但是没有在用户程序中使用的符号。单击工具栏上的“不带符号的地址”按钮，可以显示已经在用户程序中使用、但是没有在符号表中定义的绝对地址。

5.4　练习题

1．简述划分步的原则。

2．简述转换实现的条件和转换实现时应完成的操作。

3．图 5-46 是交通灯一个周期的波形图，PLC 上电后交通灯将按顺序功能图所示的要求不断地循环工作，直到 PLC 断电为止。用单序列画出顺序功能图。用时钟存储器位实现绿灯的闪烁。生成项目，用 PLCSIM 调试编写的程序，直到满足要求。

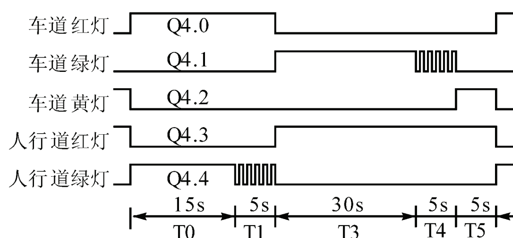


图 5-46　交通灯的波形图

4．用包含并行序列的顺序功能图来描述图 5-46 中的交通灯控制系统。生成项目，用 PLCSIM 调试编写的程序，直到满足要求。

5．液体混合装置如图 5-47 所示，上限位、下限位和中限位液位传感器被液体淹没时为 1 状态，阀 A、阀 B 和阀 C 为电磁阀，线圈通电时打开，线圈断电时关闭。开始时容器是空的，各阀门均关闭，各传感器的输出信号均为 0 状态。按下起动按钮后，将自动顺序运行。首先打开阀 A，液体 A 流入容器，中限位开关变为 1 状态时，关闭阀 A，打开阀 B，液体 B 流入容器。当液面到达上限位开关时，关闭阀 B，电机 M 开始运行，搅动液体，60s 后停止搅动，打开阀 C，放出混合液，当液面降至下限位开关之后再过 5s，容器放空，关闭阀 C，打开阀 A，又开始下一周期的操作。按下停止按钮，在当前工作周期的操作结束后，才停止操作(停在初始状态)。画出 PLC 的外部接线图和控制系统的顺序功能图，设计出梯形图程序。生成项目，用 PLCSIM 调试编写的程序，直到满足要求。

6．指出图 5-48 所示顺序功能图中的错误。

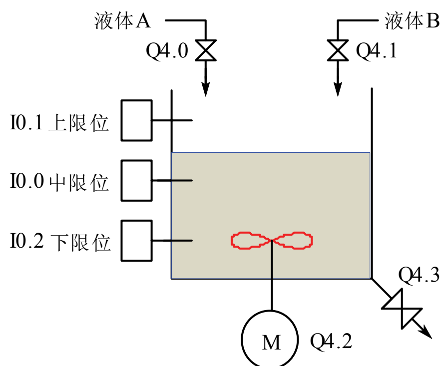


图 5-47　液体混合装置示意图

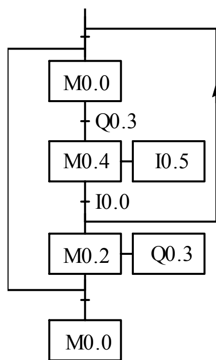


图 5-48　顺序功能图

第 6 章

网络通信

6.1 PROFIBUS-DP 网络通信

6.1.1 西门子工业通信网络简介

1．西门子工业通信网络

S7-300/400 有很强的通信功能，CPU 模块集成有 MPI 通信接口，有的 CPU 模块还集成有 PROFIBUS-DP、PROFINET 或点对点通信接口，此外还可以使用 PROFIBUS-DP、工业以太网、AS-i 和点对点通信处理器（CP）模块。通过 PROFINET、PROFIBUS-DP 或 AS-i 现场总线，CPU 与分布式 I/O 模块之间可以周期性地自动交换数据。在自动化系统之间，PLC 与计算机和 HMI（人机界面）之间，均可以交换数据。数据通信可以周期性地自动进行，或者基于事件驱动。

图 6-1 是西门子的工业自动化通信网络的示意图。

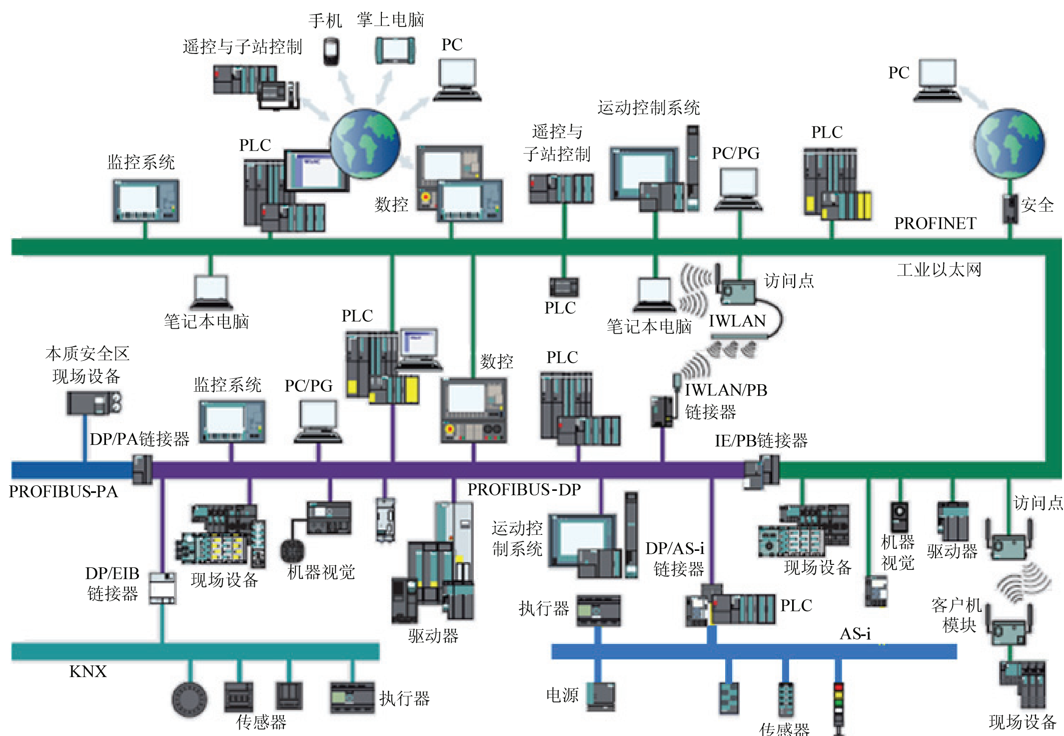


图 6-1 西门子的工业自动化通信网络

PROFINET 是基于工业以太网的现场总线，可以高速传送大量的数据。PROFIBUS 用于少量和中等数量数据的高速传送。MPI（多点接口）是 SIMATIC 产品使用的内部通信协议，用于 PLC 之间、PLC 与 HMI（人机界面）和 PG/PC（编程器/计算机）之间的通信，可以建立传送少量数据的低成本网络。点对点通信用于特殊协议的串行通信。AS-i 是底层的低成本网络，通用总线系统 KONNEX(KNX)在欧洲用于楼宇自动控制。IWLAN 是工业无线局域网的缩写。

2．PROFIBUS

西门子通信网络的中间层为工业现场总线 PROFIBUS，它是用于车间级和现场级的国际标准，传输速率最高 12Mbit/s，响应时间的典型值为 1ms，使用屏蔽双绞线电缆（最长 9.6km）或光缆（最长 90km），最多可以接 127 个从站。

PROFIBUS 是开放式的现场总线，已被纳入现场总线的国际标准 IEC 61158，并于 2006 年成为我国首个现场总线国家标准（GB/T 20540-2006）。

PROFIBUS 提供了下列 3 种通信协议：

1）PROFIBUS-FMS（现场总线报文规范）主要用于系统级和车间级的不同供应商的自动化系统之间传输数据，处理单元级（PLC 和 PC）的多主站数据通信，FMS 已基本上被以太网取代，现在很少使用。

2）PROFIBUS-DP（分布式外部设备）特别适合于 PLC 与现场级分布式 I/O 设备（例如西门子的 ET 200 和变频器）之间的通信。主站之间的通信为令牌方式，主站与从站之间为主从方式，以及这两种方式的组合。DP 是 PROFIBUS 中应用最广的通信方式。

3）PROFIBUS-PA（过程自动化）用于过程自动化的现场传感器和执行器的低速数据传输。由于传输技术采用 IEC 1158-2 标准，确保了本质安全，可以用于防爆区域的传感器和执行器与中央控制系统的通信。PROFIIBUS-PA 使用屏蔽双绞线电缆，由总线提供电源。

6.1.2 ET 200

西门子的 ET 200 是基于现场总线 PROFIBUS-DP 或 PROFINET 的分布式 I/O，可以与经过认证的非西门子公司生产的 PROFIBUS-DP 主站协同运行。

在组态时，STEP 7 自动分配标准的 DP 从站的输入/输出地址。就像访问主站主机架上的 I/O 模块一样，DP 主站的 CPU 通过 DP 从站的地址直接访问它们。因此使用标准 DP 从站不会增加编程的工作量。

1．安装在控制柜内的 ET 200

（1）ET 200S

ET 200S（见图 6-2）是一种多功能按位模块化的 I/O 系统，配备了 PROFIBUS-DP 或 PROFINET 接口模块，可以提供集成光纤接口。模块的种类丰富，有数字量 I/O 模块、模拟量 I/O 模块、技术功能模块、电动机起动器和变频器、IQ-Sense（智能传感器）模块、气动接口模块、故障安全模块。每个站最多可以使用 63 个 I/O 模块，或 20 个电动机起动器和变频器，有 2、4、8 点的 I/O 模块。能在运行时更换 I/O 模块（有热插拔功能），可以用于危险区域 Zone 2。

ET 200S COMPACT（紧凑型）有 32 点数字量 I/O，可以扩展 12 个 ET 200S 的 I/O 模块。IM 151-7 CPU 接口模块的功能与 CPU 314 相当。图 6-2 中的 ET 200S 配有 I/O 模块、电



动机起动器和变频器。

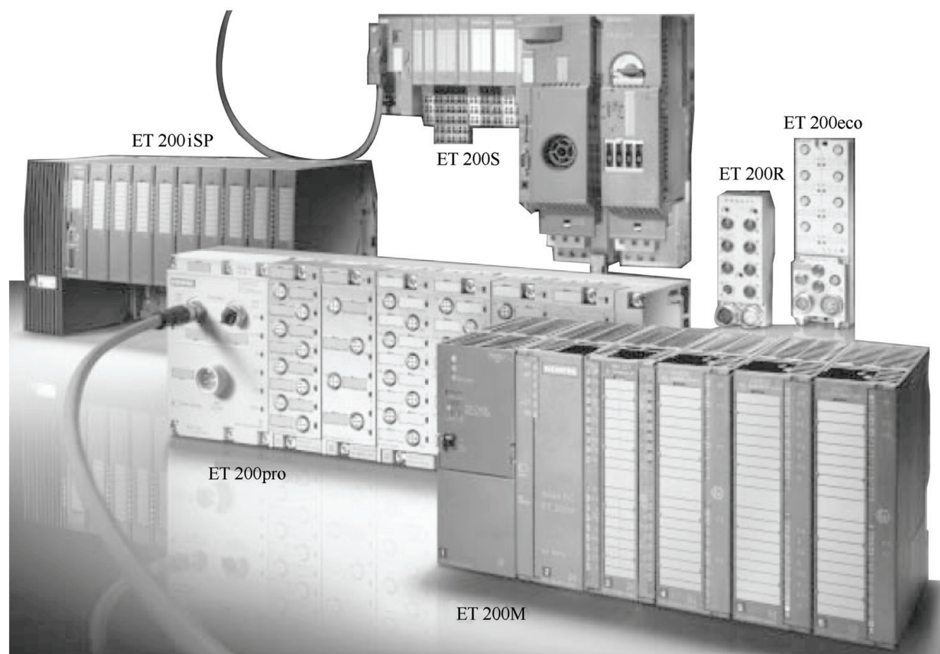


图 6-2　ET 200

（2）ET 200M

ET 200M 是多通道模块化的分布式 I/O，使用 S7-300 全系列模块，适用于大点数、高性能的应用。最多可以扩展 8 个模块，用接口模块 IM 153 来实现与主站的通信。

ET 200M 可以提供与 S7-400H 系统相连的冗余接口模块和故障安全型 I/O 模块。可以用于 Zone 2 的危险区域，传感器和执行器可以用于 Zone 1。有可以带电热插拔的模块，可在运行中修改组态。

S7-400 的 I/O 模块平均每点的价格比 S7-300 的贵得多，较大型的控制系统常用功能强大的 S7-400 的 CPU 和 ET 200M 来组成系统，这样可以使用价格便宜的 S7-300 的模块，使系统具有很高的性能价格比。这是一种常见的硬件配置方案。

ET 200M 的 SIPLUS 版本是为户外应用设计的，其环境温度范围可达−25～+60°C。

（3）ET 200iSP

ET 200iSP 是本质安全 I/O 系统，适用于有爆炸危险的区域。模块化 I/O 可以直接安装在 Zone 1，可以连接来自 Zone 0 的本质安全的传感器和执行器。

ET 200iSP 可以扩展多种端子模块，有热插拔功能，最多可以插入 32 块电子模块。ET 200iSP 也可以用于容错系统的冗余运行。ET 200iSP 有支持 HART 通信协议的模块，可以将 HART 仪表接入现场总线。

2．不需要控制柜的 ET 200

由于有很高的保护等级，具有抗冲击、防尘和不透水性，ET 200pro、ET 200eco 和 ET 200R 能适应恶劣的工业环境，可以用于没有控制柜的 I/O 系统。它们只需要很少的附加部件。

ET 200 无控制柜系统安装在一个坚固的玻璃纤维加强塑壳内，耐冲击和污物，不透水。而且附加部件少，节省布线，响应快。

（1）ET 200pro

ET 200pro 是防护等级为 IP 65/67 的多功能模块化分布式 I/O，可以直接安装在现场，有一个 DP 接口或一个 PROFINET 接口。ET 200pro 由接口模块和最多 16 个 I/O 模块组成，可以扩展数字量和模拟量 I/O 模块、电动机起动器、变频器、安全系统和气动模块等，支持故障安全功能。ET 200pro 可以热插拔和独立布线。

（2）ET 200eco

ET 200eco 是可以直接安装在现场的一体化经济实用的 I/O，能在运行时更换模块，不会中断总线或供电。

（3）ET 200R

ET 200R 是直接安装在现场的加固型 I/O，特别适合在机器人所处的恶劣的工业环境中使用，坚固的金属外壳使它能抗强电磁干扰和抗焊接火花的飞溅。ET 200R 最多有 32 个 I/O 点。

3．其他型号的 ET 200

ET 200B 是紧凑的分布式 I/O，具有不可更改的固定的输入区和输出区。提供不同电压范围和不同通道数的数字量和模拟量 I/O 模块。

6.1.3 实训三十四 组态分布式 I/O

PROFIBUS-DP 最大的优点是使用简单方便，在大多数甚至绝大多数实际应用中，只需要对网络通信作简单的组态，不用编写任何通信程序，就可以实现 DP 网络的通信。在编程时，用户程序对远程 I/O 的访问，就像访问中央机架的 I/O 一样。

实现主站与 PROFIBUS-DP 分布式 I/O 通信的主要操作是组态。

1．组态 DP 主站系统

用新建项目向导生成一个名为“ET200”的项目（见随书光盘中的同名例程），CPU 为 CPU 315-2DP（见图 6-3）。项目中的 OB82、OB86 和 OB122 中没有程序，它们的作用将在 7.1.1 节介绍。

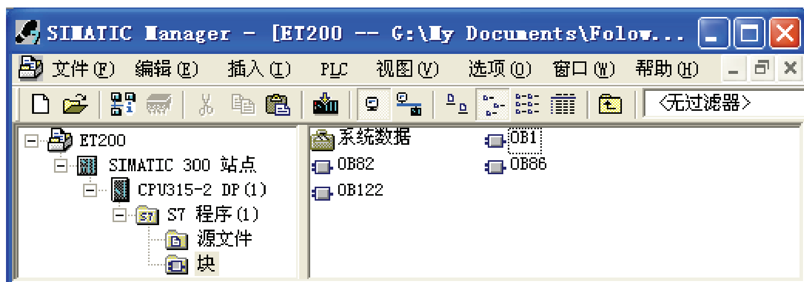


图 6-3 SIMATIC 管理器

选中 SIMATIC 管理器左边窗口出现的“SIMATIC 300 站点”，双击右边窗口中的“硬件”图标，打开硬件组态工具 HW Config（见图 6-4）。可以看到自动生成的机架和 2 号槽中的 CPU 模块。将电源模块插入 1 号槽，16 点数字量输入模块插入 4 号槽，16 点数字量输出



模块插入 5 号槽。它们分别占用 IW0 和 QW4。

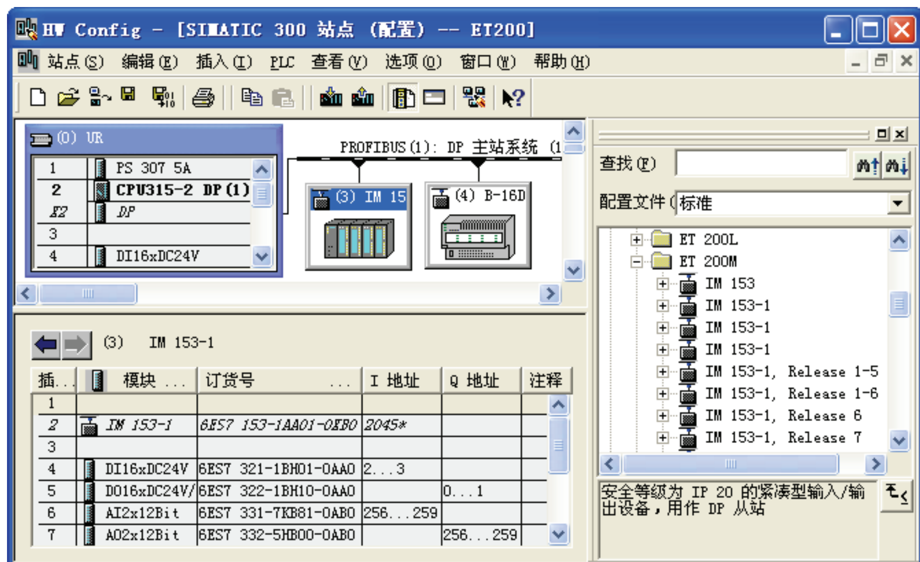


图 6-4　组态 DP 从站

用鼠标双击机架中 CPU 315-2DP 下面“DP”所在的行（见图 6-4），单击出现的 DP 属性对话框的“常规”选项卡中的“属性”按钮（见图 6-5），在出现的“属性 - PROFIBUS 接口 DP”对话框中，可以设置 CPU 在 DP 网络中的站地址，默认的站地址为 2。

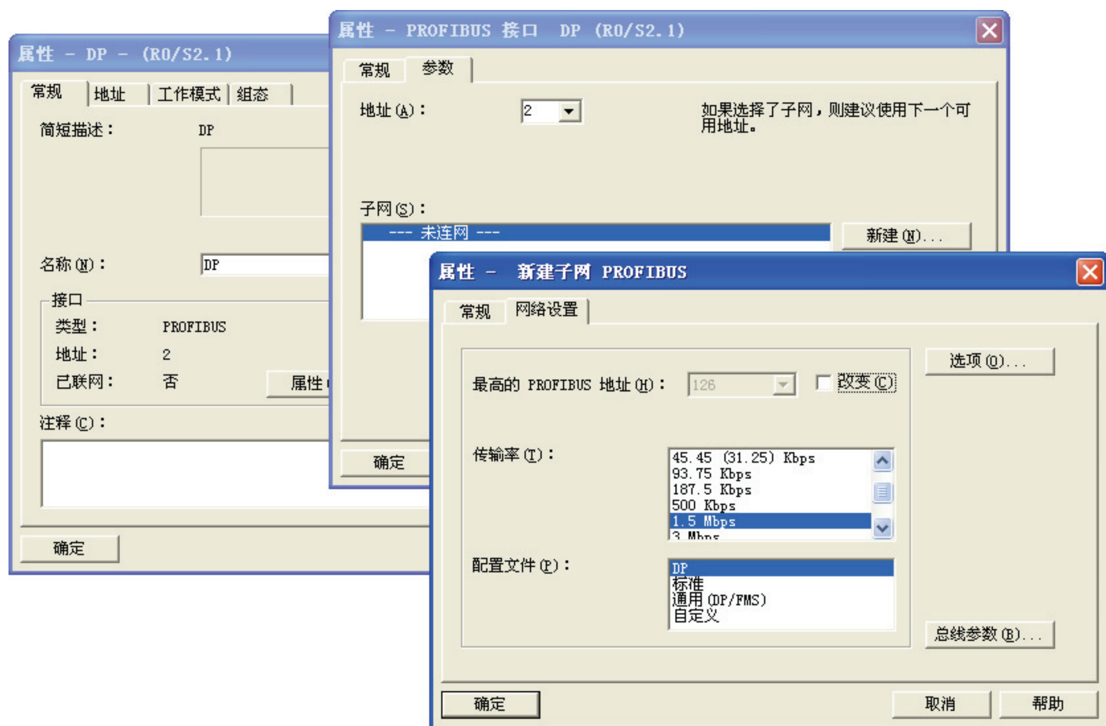


图 6-5　PROFIBUS 接口属性对话框

单击“新建”按钮，在出现的“属性 – 新建子网 PROFIBUS”对话框的“网络设置”选项卡中（见图 6-5），采用系统推荐的默认参数：传输速率为 1.5 Mbit/s，配置文件为 DP。传输速率和总线配置文件将用于整个 PROFIBUS 子网络。

单击“新建子网”对话框中的“确定”按钮，返回 PROFIBUS 接口属性对话框（见图 6-5）。可以看到“子网”列表框中出现了新生成的名为“PROFIBUS（1）”的子网。两次单击“确定”按钮，返回 HW Config，此时只能看到 S7-300 的机架和新生成的 PROFIBUS（1）网络线。图 6-4 是已经组态好的 PROFIBUS 网络和从站。

2．组态 ET 200M 从站

打开硬件目录窗口的文件夹“\PROFIBUS-DP\ET 200M”，将其中的接口模块 IM 153-1（见图 6-4 的右边窗口）拖放到 PROFIBUS 网络线上，就生成了 ET 200M 从站。在出现的“属性–PROFIBUS 接口 IM 153-1”对话框中，设置它的站地址为 3。用 IM 153-1 模块上的 DIP 开关（见图 6-6）设置的站地址应与 STEP 7 组态的站地址相同。

选中图 6-4 上面窗口的该从站，下面窗口是它的机架中的槽位，其中的 4～11 号槽最多可以插入 8 块 S7-300 系列的模块。打开硬件目录中的“IM 153-1”子文件夹，它里面的各子文件夹列出了可用的 S7-300 模块，其组态方法与普通的 S7-300 的相同。将 DI、DO、AI、AO 模块分别插入 4~7 号槽。

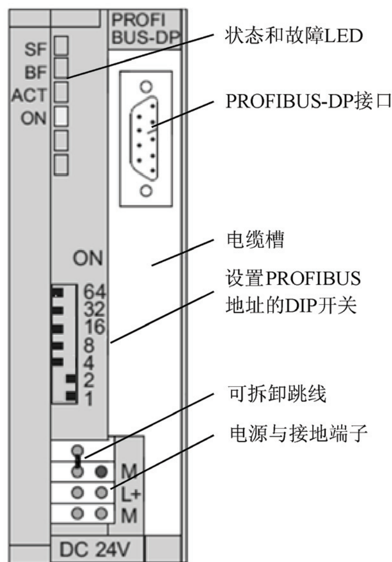


图 6-6　IM 153-1 的正面视图

打开硬件目录窗口的文件夹“\PROFIBUS-DP\ET 200B”，将其中的 B-16DI/16DO 拖放到 PROFIBUS 网络线上，就生成了 ET 200B 从站。在出现的“属性 – PROFIBUS 接口 B-16DI/16DO”对话框中，自动设置的站地址为 4。用模块上的拨码开关设置的站地址应与 STEP 7 组态的站地址相同。

3．DP 网络上的 I/O 地址分配

在 PROFIBUS 网络系统中，主站和非智能从站的 I/O 自动统一分配地址。下面是模块地址分配的原则：

1）I/O 分为 4 类，即数字量输入、数字量输出、模拟量输入和模拟量输出。按组态的先后次序，同类 I/O 模块的字节地址依次顺序排列。

2）数字量 I/O 模块的起始地址从 0 号字节开始分配。S7-300 和 S7-400 的模拟量 I/O 模块的起始地址分别从 256 号和 512 号字节开始分配，每个模拟量 I/O 点占 2B（两个字节）的地址。模块地址与模块所在的机架号和插槽号无关。

4．通信的仿真验证

组态任务完成后，单击工具栏上的按钮，首先对组态信息进行编译。如果组态存在问题，将会显示错误或警告信息。编译成功后，组态信息保存在系统数据中。组态信息下载到 CPU 后起作用。

为了验证 CPU 与 DP 从站之间的通信，在主程序 OB1 中编写下面的简单程序：



L　　IW　　2
T　　QW　　0

即用 ET 200M 的数字量输入来控制它的数字量输出。

打开 PLCSIM，生成 IB2 和 QB0 的视图对象（见图 6-7）。将系统数据和 OB1 下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。

用鼠标将 IB2 中的某些位设置为 1（小方框内出现“√”），由于主站与从站之间的通信和 OB1 中的程序，QB0 中的对应位变为 1 状态。



图 6-7　PLCSIM

5．ET 200S 的组态与仿真练习

用新建项目向导生成一个项目，CPU 为 CPU 316-2DP。

打开硬件组态工具 HW Config，插入电源模块和信号模块。生成 DP 网络，采用默认的参数。打开硬件目录窗口的文件夹“\PROFIBUS-DP\ET 200S”，将其中的接口模块 IM 151-1 Basic 拖放到 PROFIBUS 网络线上，生成 ET 200S 从站，设置它的站地址为 3。选中生成的从站，打开硬件目录中的子文件夹“\IM 151-1 Basic\PM”，将其中的 AC 电源模块插入 1 号槽。将子文件夹“\IM 151-1 Basic\DI”中的 4 块“4DI DC24V ST”模块插入 2~5 号槽（见图 6-8）。

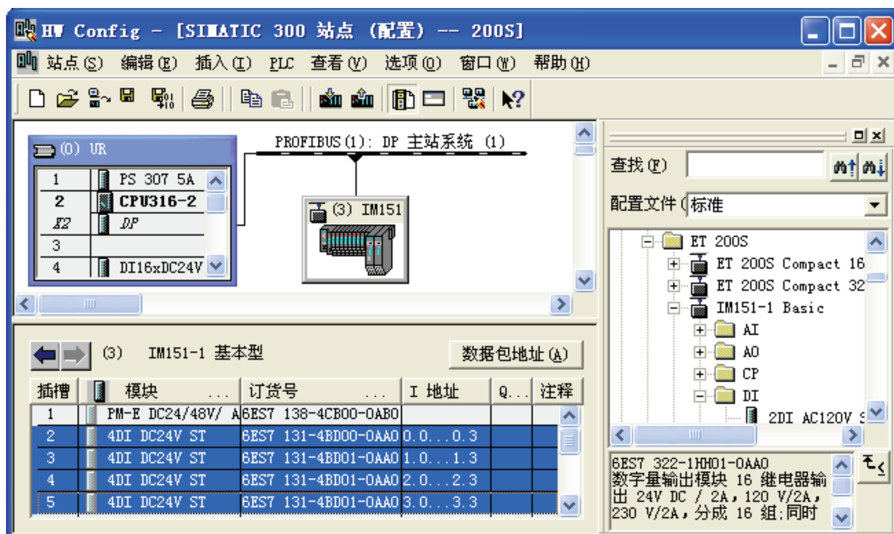


图 6-8　组态 DP 从站

可以看到各 DI 模块被分配了一个字节的地址，但是只使用了其中的 4 位，相邻 DI 模块

的地址不是连续的。对于相邻的只有 2 点的 DI 模块或 DO 模块来说，这个问题更为严重。可以用下面的方法使地址连续。

按住计算机的〈Ctrl〉键，单击下面的 ET 200S 的“插槽”列的 2~5 号槽，选中它们后，其背景色变为深蓝色（见图 6-8）。单击“数据包地址”按钮，可以看到 4 个 DI 模块的地址被自动调整，只占 2B 了。将子文件夹“\IM 151-1 Basic\DO”中的 4 块“4DO DC24V ST”模块插入 6~9 号槽。用上述的方法修改模块的地址，使之连续。保存和编译组态信息。

在主程序 OB1 中编写程序，用 ET 200S 的数字量输入来控制它的数字量输出。

打开 PLCSIM，将系统数据和 OB1 下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。检查是否能用 ET 200S 的数字量输入来控制它的数字量输出。

6.1.4　实训三十五　组态 DP 主站与 S7-200 的通信

PROFIBUS-DP 是通用的国际标准，符合该标准的第三方设备作 DP 网络的从站时，需要在 HW Config 中安装 GSD 文件，才能在硬件目录窗口看到第三方设备和对它进行组态。本例程组态 DP 主站与 S7-200 的 PROFIBUS 通信。

1. PROFIBUS-DP 从站模块 EM 277

DP 从站模块 EM 277 用于将 S7-200 CPU 连接到 DP 网络，波特率为 9.6k～12Mbit/s。主站可以读写 S7-200 的 V 存储区，每次可以与 EM 277 交换 1～128 个字节的数据。EM 277 只能作 DP 从站，不需要在 S7-200 一侧对 DP 通信组态和编程。

2. 组态 S7-300 站

用新建项目向导生成一个名为“EM277”的项目（见随书光盘中的同名例程），CPU 为 CPU 315-2DP。

选中 SIMATIC 管理器左边窗口出现的“SIMATIC 300 站点”，双击右边窗口中的“硬件”图标，打开硬件组态工具 HW Config（见图 6-9）。可以看到自动生成的机架和 2 号槽中的 CPU 模块。将电源模块插入 1 号槽，16 点 DI 模块插入 4 号槽，16 点 DO 模块插入 5 号槽。它们分别占用 IW0 和 QW4。

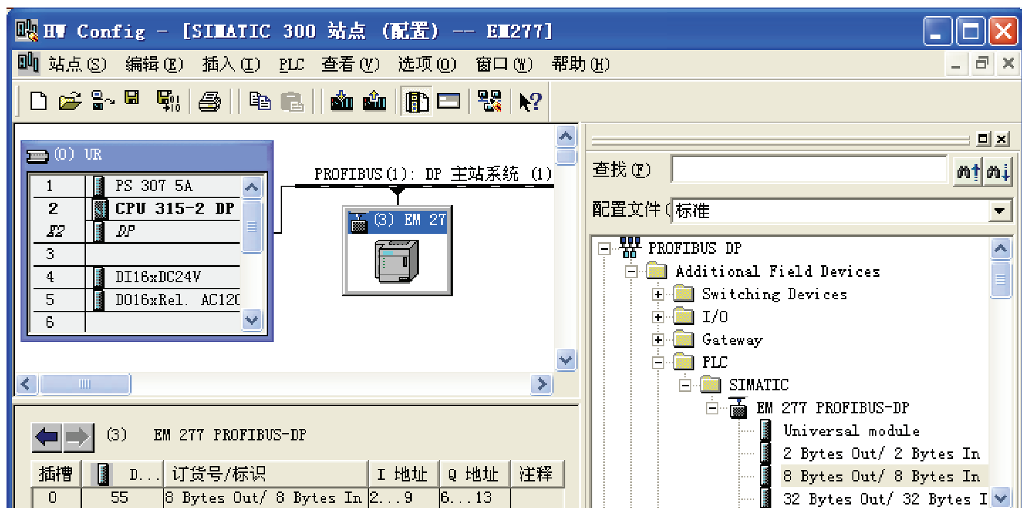


图 6-9　组态 DP 从站



双击“DP”所在的行，单击打开的对话框的“常规”选项卡中的“属性”按钮（见图 6-5），在出现的对话框的“参数”选项卡中，单击“新建”按钮，生成一条 PROFIBUS-DP 网络，采用默认的网络参数和默认的站地址 2。单击 3 次“确定”按钮，返回 HW Config。

3．安装 EM 277 的 GSD 文件

EM 277 作为 PROFIBUS-DP 从站模块，其有关参数是以 GSD 文件的形式保存的。在对 EM 277 组态之前，需要安装它的 GSD 文件。EM 277 的 GSD 文件 siem089d.gsd 在随书光盘的文件夹“\Project”中。

执行 HW Config 中的菜单命令“选项”→“安装 GSD 文件”，在出现的“安装 GSD 文件”对话框中（见图 6-10），用最上面的选择框选中 GSD 文件“来自目录”。单击“浏览”按钮，用出现的“浏览文件夹”对话框选中随书光盘中的文件夹“\Project”，单击“确定”按钮，该文件夹的 GSD 文件“siem089d.gsd”出现在 GSD 文件列表框中。选中需要安装的 GSD 文件，单击“安装”按钮，开始安装。

安装结束后，在 HW Config 右边的硬件目录窗口的“\PROFIBUS DP\Additional Field Devices\PLC\SIMATIC”文件夹中，可以看到新安装的 EM 277（见图 6-9）。

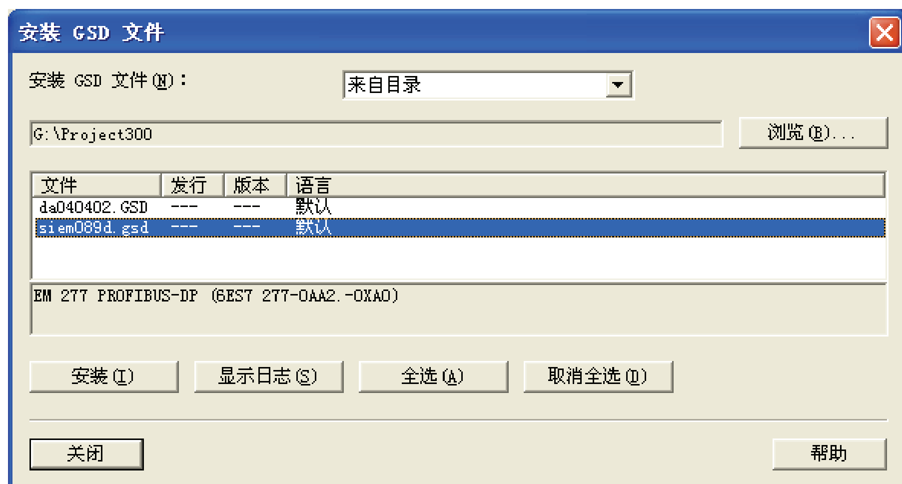


图 6-10　安装 GSD 文件

4．不能安装 GSD 的处理方法

随书光盘中的项目“EM 277”带有 EM 277 的 GSD 文件，打开该项目后，在 HW Config 的硬件窗口看不到 EM 277。如果安装来自目录的 GSD 文件，将会出现无法安装的警告信息。这是因为打开该项目时，EM 277 的 GSD 文件被引用。必须关闭有 DP 从站的项目，重新启动计算机，打开别的项目，才能安装 EM 277 的 GSD 文件。

也可以打开别的项目，用选择框选中“来自项目”，安装项目“EM277”中的 GSD 文件。

5．组态 EM 277 从站

安装 GSD 文件后，将 HW Config 右侧窗口的“EM 277 PROFIBUS-DP”拖放到左边窗口的 PROFIBUS-DP 网络上。用鼠标选中生成的 EM 277 从站，打开右边窗口的设备列表中的“\EM 277 PROFIBUS-DP”子文件夹，根据实际系统的需要选择传送的通信字节

数。例程 EM277 选择的是 8 字节输入/8 字节输出方式，将图 6-9 中的“8 Bytes Out/ 8 Bytes In”拖放到下面窗口的表格中的 1 号槽。STEP 7 自动分配远程 I/O 的输入/输出地址，因为主机架占用了 IW0 和 QW4，分配给 EM 277 模块的输入、输出地址分别为 IB2~IB9 和 QB6~QB13。

双击网络上的 EM 277 从站，打开 DP 从站属性对话框。单击“常规”选项卡中的“PROFIBUS…”按钮，在打开的接口属性对话框中，设置 EM 277 的站地址为 3。用 EM 277 上的拨码开关设置的站地址应与 STEP 7 中设置的站地址相同。

在“参数赋值”选项卡中（见图 6-11），设置“I/O Offset in the V-memory”（V 存储区中的 I/O 偏移量）为 100，即用 S7-200 的 VB100～VB115 与 S7-300 的 QB6~QB13 和 IB2~IB9 交换数据。组态结束后，应将组态信息下载到 S7-300 的 CPU 模块。

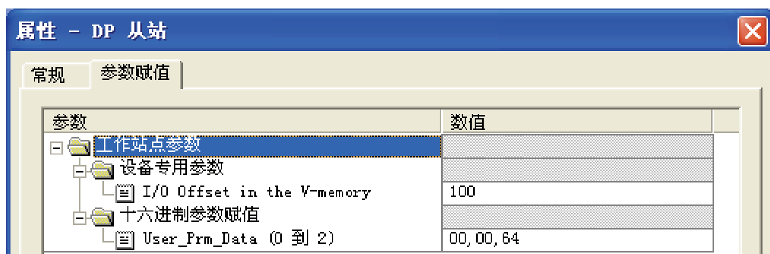


图 6-11　DP 从站属性对话框

6．S7-200 的编程

本例的 S7-200 通过 VB100～VB115 与 DP 主站交换数据。S7-300 写到 S7-200 的数据保存在 VB100～VB107，对应于 S7-300 的 QB6~QB13； S7-300 从 S7-200 的 VB108～VB115 读取数据，对应于 S7-300 的 IB2~IB9。

如果要把 S7-200 的 MB3 的值传送给 S7-300 的 MB10，应在 S7-200 的程序中，用 MOVB 指令将 MB3 传送到 VB108～VB115 中的某个字节，例如 VB108。通过通信，VB108 的值传送给 S7-300 的 IB2，在 S7-300 的程序中将 IB2 的值传送给 MB10。

7．变频器作 DP 从站的组态练习

变频器安装一块 PROFIBUS-DP 接口板后，可以作 DP 网络中的从站，主站可以监控多台变频器。在 STEP 7 的 HW Config 的硬件目录窗口的“PROFIBUS DP”文件夹中，有西门子的各种变频器和直流调速装置，可以用它们来组态驱动器从站。

下面的练习首先安装丹麦丹佛斯公司 VLT 5000 系列的变频器的 GSD 文件，然后组态变频器从站。

用新建项目向导生成一个项目，CPU 为 CPU 315-2DP。打开硬件组态工具 HW Config，插入电源模块和信号模块。生成 DP 网络，采用默认的网络参数。

丹佛斯 VLT 5000 系列变频器的 GSD 文件 da040402.GSD 在随书光盘的文件夹“\Project”中。用本实训介绍的方法安装 VLT 5000 的 GSD 文件。

安装结束后，打开 HW Config 右边的硬件目录窗口的“PROFIBUS DP”文件夹，将 VLT 5000 拖放到 DP 网络上，设置它为 3 号站（见图 6-12）。选中该从站后，将子文件夹“VLT 5000”中的“PPO Typ 1 Module consisitent PCD”拖放到下面窗口的第一行。下面窗口出现两行，第 1 行为 4 个字的参数数据，第 2 行为两个字的过程数据。从站的输入、输出数



据的意义和通信程序的设计、调试方法，见作者编写的《西门子工业通信网络组态编程与故障诊断》的 3.3 节。

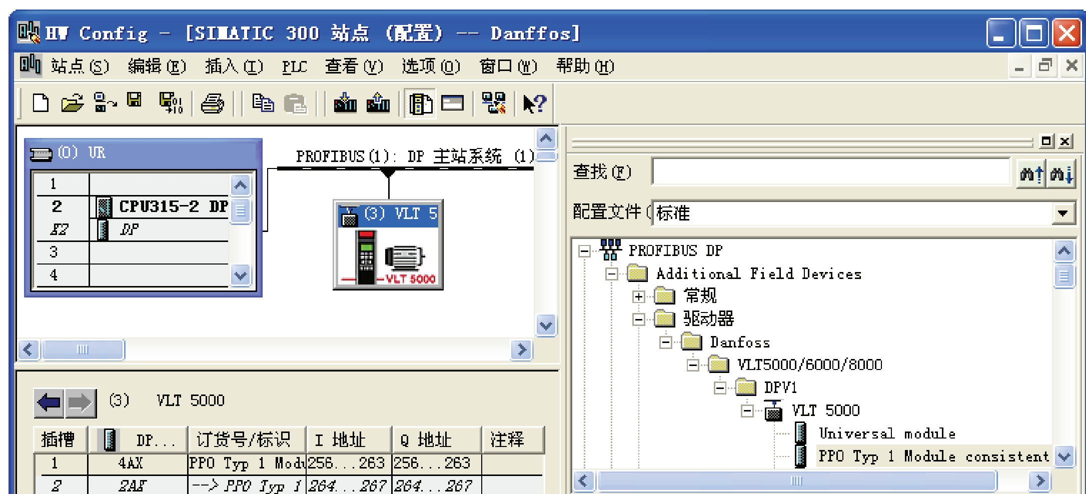


图 6-12　组态 DP 从站

6.1.5　实训三十六　组态 DP 主站与智能从站的主从通信

可以将自动化任务划分为用多台 PLC 控制的若干个子任务，这些子任务分别用几台 CPU 独立地和有效地进行处理，这些 CPU 在 DP 网络中作 DP 主站和智能从站。

主站和智能从站的地址是独立的，它们可能分别使用相同的 I/O 地址区。DP 主站不是用 I/O 地址直接访问智能从站的物理 I/O 区，而是通过从站组态时指定的通信双方的 I/O 区来交换数据。该 I/O 区不能占用分配给 I/O 模块的物理 I/O 地址区。

主站与从站之间的数据交换是由 PLC 的操作系统周期性自动完成的，不需要用户编程，但是用户必须对主站和智能从站之间的通信连接和用于数据交换的地址区组态。这种通信方式称为主/从（Master/Slave）通信方式，简称为 MS 方式。

1．组态 DP 主站和 PROFIBUS 网络

用新建项目向导生成一个名为“智能从站”的项目（见随书光盘中的同名例程），CPU 为 CPU 412-2DP。

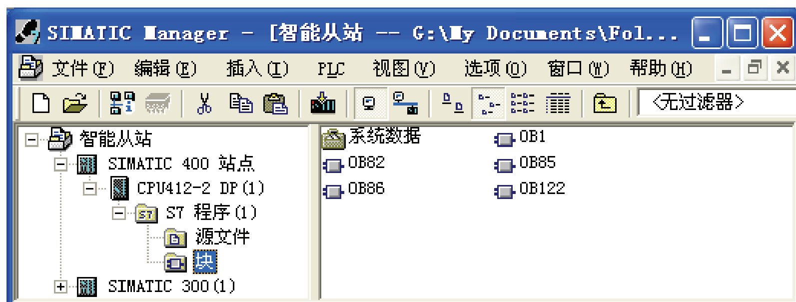


图 6-13　SIMATIC 管理器

选中 SIMATIC 管理器左边窗口出现的“SIMATIC 400 站点”（见图 6-13），双击右边窗口中的“硬件”图标，打开硬件组态工具 HW Config（见图 6-14）。将电源模块插入 1 号槽，将 16 点 DI 模块插入 5 号槽，16 点 DO 模块插入 6 号槽。它们的地址分别为 IW0 和 QW0。

双击机架中 CPU 412-2DP 下面“DP”所在的行，打开 PROFIBUS 接口属性对话框。单击“常规”选项卡中的“属性”按钮（见图 6-5），在出现的对话框的“参数”选项卡中，单击“新建”按钮，生成一条 PROFIBUS-DP 网络，采用默认的网络参数和默认的站地址 2。单击 3 次“确定”按钮，返回 HW Config。可以看到生成的 PROFIBUS 网络线，此时还没有图 6-14 中的 3 号从站。

单击工具栏上的按钮，编译与保存组态信息。最后关闭 HW Config，返回 SIMATIC 管理器。

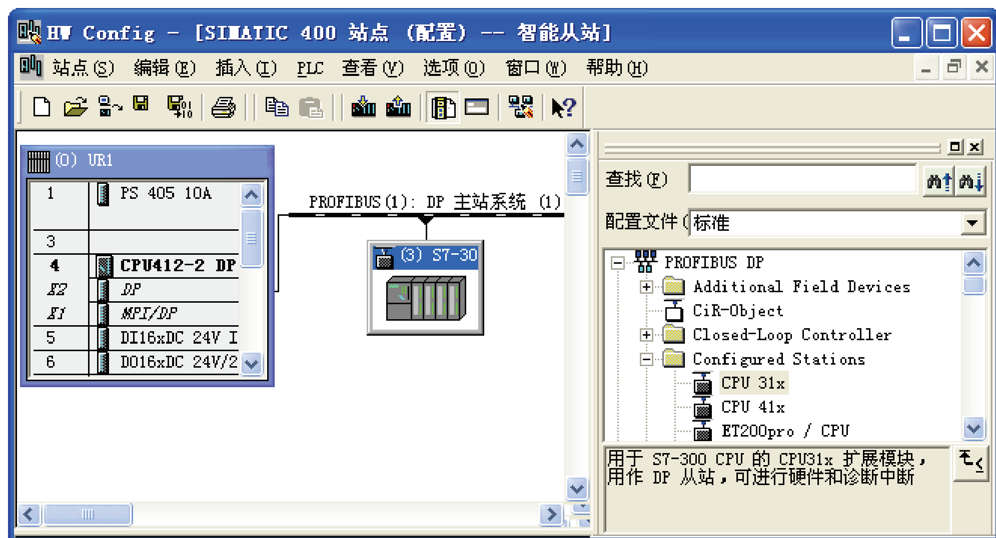


图 6-14　组态 PROFIBUS 智能从站

2. 组态智能从站

用鼠标右键单击 SIMATIC 管理器左边窗口最上面的项目图标，执行出现的快捷菜单中的命令“插入新对象”→“SIMATIC 300 站点”。选中左边窗口新出现的“SIMATIC 300(1)”，用鼠标左键双击右边窗口的“硬件”图标，打开 HW Config。将硬件目录窗口的文件夹\SIMATIC 300\RACK-300 中的导轨（Rail）拖放到硬件组态窗口。

将硬件目录窗口的文件夹“\SIMATIC 300\PS 300”中的“PS 307 5A”插入 1 号槽。将 CPU 315-2DP 插入 2 号槽。CPU 插入机架时，将会自动打开 DP 接口属性对话框的“参数”选项卡。设置 PROFIBUS 站地址为 3，不连接到 PROFIBUS(1)网络。将 CPU 的 MPI 地址设置为 3，单击“确定”按钮返回 HW Config。

打开硬件目录窗口中的文件夹\SIMATIC 300\SM-300，将 16 点 DI 模块插入 4 号槽，16 点 DO 模块插入 5 号槽。它们分别占用 IW0 和 QW4。

双击机架中 CPU 315-2DP 下面的“DP”所在的行，打开 DP 属性对话框。在“工作模



式”选项卡将该站设置为 DP 从站（见图 6-15），单击“确定”按钮返回 HW Config。

不是所有的 CPU 都能作 DP 从站，具体的情况可以查阅有关的手册或产品样本。在 HW Config 的硬件目录窗口下面的小窗口中，可以看到对选中的硬件的简要介绍。

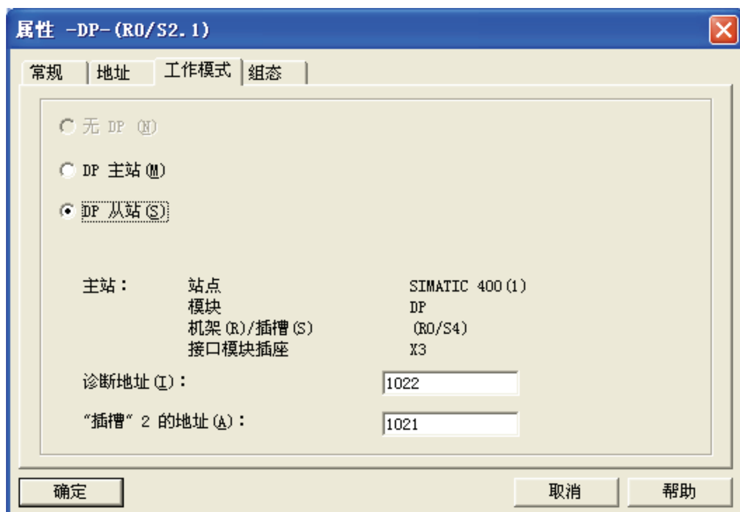


图 6-15　组态智能从站的工作模式

因为此时从站与主站的通信组态还没有结束，不能成功地编译 S7-300 的硬件组态信息。单击按钮，保存组态信息。最后关闭 HW Config。

3. 将智能 DP 从站连接到 DP 主站系统

选中 SIMATIC 管理器中的 S7-400 站，双击右边窗口的“硬件”图标，打开 HW Config。打开右边的硬件目录窗口中的“\PROFIBUS DP\Configured Stations”（已组态的站）文件夹（见图 6-14），将其中的“CPU 31x”拖放到屏幕左上方的 PROFIBUS 网络线上。“DP 从站属性”对话框的“连接”选项卡（见图 6-16）被自动打开，选中从站列表中的“CPU 313-2DP”，单击“连接”按钮，该从站被连接到 DP 网络上。



图 6-16　连接智能从站

连接好从站后，单击“确定”按钮，关闭 DP 从站属性对话框，返回 HW Config。单击“断开连接”按钮，可以将从站从网络上断开。

4．主站与智能从站主从通信的组态

用鼠标双击已连接到 PROFIBUS 网络上的 DP 从站，打开 DP 从站属性对话框的“组态”选项卡（见图 6-17 上面的图），为主-从通信设置双方用于通信的输入/输出地址区。

单击图中的“编辑”按钮，可以编辑选中的行。单击“删除”按钮，将删除选中的行。

图 6-17 中的模式“MS”表示主从通信；伙伴（主站）地址和本地（从站）地址是输入/输出地址区的起始地址，“长度”的单位可以选字节和字。数据的“一致性”的意义见实训三十七。

单击“新建”按钮，在出现的对话框中（见图 6-17 下面的图）设置组态表第 1 行的参数。每次可以设置智能从站与主站一个方向的通信使用的 I/O 地址区。设置好以后单击“确定”按钮，返回 DP 从站属性对话框的“组态”选项卡。

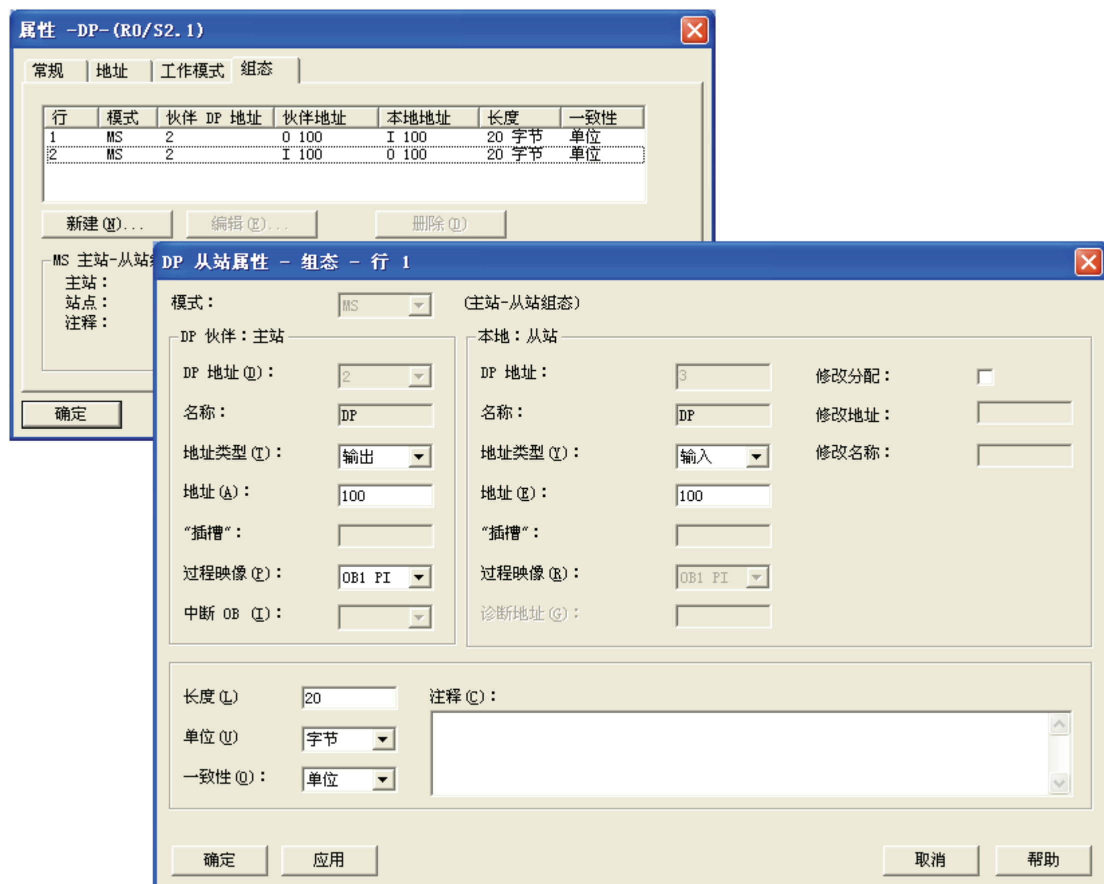


图 6-17 组态 DP 主从通信的输入/输出地址区

“组态”选项卡的第 1 行表示从站的通信伙伴（即主站）用 QB100～QB119 发送数据给从站（本地）的 IB100～IB119。第 2 行表示主站用 IB100～IB119 接收从站的 QB100～QB119 发送给它的数据。每一行最多能组态 32B，如果超出允许的字节数，单击“确定”按



钮时将会出现提示信息。

组态第 2 行的通信参数时，将“DP 从站属性 － 组态 － 行 2”对话框中的 DP 伙伴（主站）的“地址类型”改为“输入”，本地（从站）的地址类型自动变为“输出”。其余的参数与图 6-17 中的相同。

图 6-17 中组态的通信双方使用的输入/输出区的起始字节地址均为 100（IB100 和 QB100），并不要求一定要将它们设置得相同。但是用于通信的数据区不能与实际硬件占用的输入/输出区（包括主站和非智能从站的输入/输出区）重叠。

设置好全部参数后，返回 HW Config，单击工具栏上的按钮，编译与保存 SIMATIC 400 站点的组态信息。返回 SIMATIC 管理器后，选中 SIMATIC 300（1）站点，打开 HW Config。因为已完成了所有的组态任务，单击工具栏上的按钮，可以成功地编译与保存组态信息。

单击工具栏上的按钮，打开网络组态工具 NetPro（见图 6-18），可以看到两个站点都连接到 PROFIBUS 网络上。

单击站点左边方框中的 PLC 图标，可以打开 HW Config，为该站点的硬件组态。

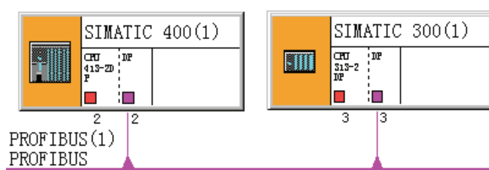


图 6-18 NetPro

5．通信的验证

下面是 CPU 412-2DP 的 OB1 中的程序，CPU 315-2DP 的 OB1 的程序基本上相同，只是将 QW0 改为 QW4。

```
L    IW    0          //IW0 的值送发送缓冲区的 QW100
T    QW    100
L    IW    100        //用 IW100 接收到的数据控制 QW0
T    QW    0
```

DP 与智能从站的通信不能用 PLCSIM 来仿真，只能用硬件来验证。

将通信双方的程序块和组态信息下载到 CPU。用 PROFIBUS 电缆连接主站和从站的 DP 接口，接通主站和从站的电源。将 CPU 切换到 RUN 模式，通过通信，可以用双方的 IW0 控制对方的 QW0 或 QW4。

6．DP 智能从站的组态练习

组态一个项目，CPU 315-2DP 为 DP 主站，CPU 313C-2DP 为智能从站，双方分别用 IB60 和 QB60 开始的输入、输出地址区双向传送 10B 的数据，一致性为“单位”。

6.1.6 实训三十七 用 SFC14 和 SFC15 传输一致性数据

1．数据的一致性

数据的一致性（Consistency）又称为连续性。通信块被执行、通信数据被传送的过程如果被一个更高优先级的 OB 块中断，将会使传送的数据不一致(不连续)。即被传输的数据一部分来自中断之前，一部分来自中断之后，因此这些数据是不连续的。

在通信中，有的从站用来实现复杂的控制功能，例如模拟量闭环控制或电气传动等。从站与主站之间需要同步传送比字节、字和双字更大的数据区，这样的数据称为一致性数据。需要绝对一致性传送的数据量越大，系统的中断反应时间越长。PROFIBUS 网络控制系统经

常使用系统功能 SFC14 和 SFC15 来传送具有一致性的数据。

2．组态硬件和主从通信

在 STEP 7 中生成一个项目（见随书光盘中的例程 SFC14_15），CPU 412-2DP 是 S7 DP 主站，CPU 313C-2DP 是智能 DP 从站。主站和从站的组态与实训三十六的项目“智能从站”基本上相同，数据长度为 20B。其区别在于参数“一致性”被组态为“全部”（见图 6-19），因此需要在用户程序中调用 SFC15“DPWR_DAT”，将数据“打包”后发送，调用 SFC14“DPRD_DAT”，将接收到的数据“解包”。SFC14、SFC15 的参数 RECORD 指定的地址区应与组态的参数一致。

DP 从站属性

常规 | 连接 | 组态

行	模式	伙伴 DP 地址	伙伴地址	本地地址	长度	一致性
1	MS	2	O 100	I 100	20 字节	全部
2	MS	2	I 100	O 100	20 字节	全部

图 6-19　主从通信组态表

DP 主站用 SFC15 发送的输出数据被智能从站用 SFC14 读出，并作为其输入数据保存。反之也适用于智能从站发送给主站的数据的处理。用于通信的输入/输出数据区的起始地址 LADDR 应使用十六进制数格式。100 对应的十六进制数为 16#64。

3．生成数据块

选中 SIMATIC 管理器左边窗口中 CPU 412-2DP 的“块”文件夹，用鼠标右键单击右边的窗口，执行出现的快捷菜单中的命令，生成数据块 DB1。打开 DB1，生成一个名为 ARAY，有 20 个字节元素的数组（见图 6-20）。

用同样的方法生成数据块 DB2，在数据块中创建一个名为 ARAY，有 20 个字节元素的数组。也可以用复制和修改名称的方法来创建内部结构相同的数据块。

LAD/STL/FBD - [DB1 -- PB_MS_3\SIMATIC 400(1)\CPU 413...

文件(F) 编辑(E) 插入(I) PLC 调试(D) 视图(V) 选项(O) 窗口(W) 帮助(H)

地址	名称	类型	初始值	注释
0.0		STRUCT		
+0.0	ARAY	ARRAY[1..20]		
*1.0		BYTE		
=20.0		END_STRUCT		

图 6-20　在数据块中生成数组

4．OB1 的程序

在双方的主程序 OB1 中，调用 SFC15“DPWR_DAT”，将 DB1 中的数据“打包”后发送，调用 SFC14“DPRD_DAT”，将接收到的数据“解包”后存放到 DB2。

输入程序时，将程序编辑器左边窗口的文件夹“\库\Standard Library\System Function Blocks”中的 SFC14“拖放”到右边窗口的程序段中，将会自动生成调用 SFC14 的 CALL 指令。下面是主站 OB1 的程序：



```
程序段 1：解开 IB100～IB119 接收到的数据包，并将数据存放在 DB2
          CALL    "DPRD_DAT"              //调用 SFC14
           LADDR    :=W#16#64             //接收通信数据的过程映像输入区的起始地址 IB100
           RET_VAL:=MW2                   //错误代码
           RECORD   :=DB2.ARAY            //存放接收的用户数据的目的数据区
程序段 2：将 DB1 的数据打包后通过 QB100～QB119 发送出去
          CALL    "DPWR_DAT"              //调用 SFC15
           LADDR    :=W#16#64             //发送数据的过程映像输出区的起始地址 QB100
           RECORD:=P#DB1.ARAY             //存放要发送的用户数据的源数据区
           RET_VAL:=MW4                   //错误代码
```

因为 DB2 中的数组的大小刚好为 20B，输入 RECORD 的实参 P#DB2.DBX0.0 BYTE 20 后，自动变为 DB2.ARAY。当然也可以直接输入 DB2.ARAY。

从站 OB1 中的程序与主站的基本上相同，图 6-21 给出了通信双方的信号关系图。

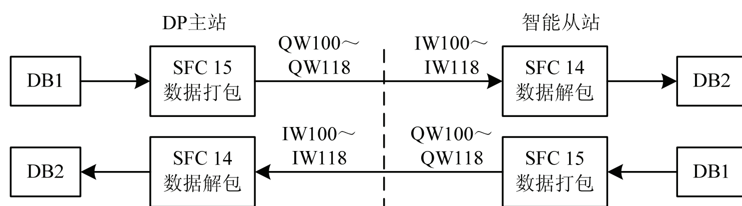


图 6-21　DP 主站与智能从站的数据传输

5．初始化程序

在主站和从站的初始化程序 OB100 中，用 SFC21 将 DB1 的数据发送区中的各个字分别预置为 16#1111 和 16#2222。将 DB2 的数据接收区中的各个字清零。

下面是 CPU 412-2DP 的 OB100 中的程序：

```
程序段 1：初始化存放要发送的数据的地址区
          L       W#16#1111
          T       LW      20
          CALL    "FILL"                        //调用 SFC21
           BVAL      :=LW20                     //源数据
           RET_VAL:=LW22                        //故障代码
           BLK       :=P#DB1.ARAY               //地址区
程序段 2：将存放接收的数据的地址区清零
          L       W#16#0
          T       LW      20
          CALL    "FILL"
           BVAL      :=LW20                     //源数据
           RET_VAL:=LW22                        //故障代码
           BLK       :=P#DB2. ARAY              //地址区
```

6．通信的验证

将通信双方的程序块和组态信息下载到硬件 CPU，用 PROFIBUS 电缆连接主站和从站的 DP 接口，接通主站和从站的电源，将 CPU 切换到 RUN 模式。

运行时在 SIMATIC 管理器中双击打开 DB2，单击工具栏上的按钮，起动监控功能。查看通信双方 DB1 中初始化的数据是否传送给了通信伙伴的 DB2。

6.2 S7 通信的组态与编程

6.2.1 S7 通信简介

1．连接的基本概念

数据通信协议可以分为面向连接的协议和无连接的协议，前者在进行数据交换之前，必须与通信伙伴建立连接。面向连接的协议具有较高的安全性。

连接是指两个通信伙伴之间为了执行通信服务建立的逻辑链路，而不是指两个站之间用物理媒体（例如电缆）实现的连接。连接相当于通信伙伴之间一条虚拟的“专线”，它们随时可以用这条“专线”进行通信。一条物理线路可以建立多个连接。

S7 连接属于需要组态的静态连接，CPU 同时可以使用的连接的个数与它们的型号有关。

基于连接的通信分为单向通信和双向通信。在双向通信中，通信双方都需要调用通信块，一方调用发送块来发送数据，另一方调用接收块来接收数据。

单向通信只需要通信的一方编写通信程序。编写通信程序一方的 CPU 为客户机（Client），不需编写通信程序一方的 CPU 为服务器（Server）。客户机是向服务器请求服务的设备，它是主动的，需要调用通信块对服务器的数据进行读、写操作。服务器是提供特定服务的设备，是通信中的被动方，通信功能由它的操作系统执行。通信服务经客户机要求而启动。

2．S7 通信

S7 通信主要用于西门子工控产品之间的通信，例如 S7-300/400 CPU 之间的主-主通信、CPU 与人机界面和组态软件 WinCC 之间的通信。S7 通信可以用于工业以太网、PROFIBUS 或 MPI 网络。这些网络的 S7 通信的组态和编程方法基本上相同。

用于数据交换的 S7 通信的 SFB/FB 见表 6-1。在 S7 单向连接中，客户机调用单向通信功能块 GET 和 PUT，读、写服务器的存储区。双向 S7 通信需要调用 SFB8/SFB9 和 SFB12/SFB13。

表 6-1　用于 S7 通信数据交换的 SFB/FB

编　号		助记符	可传输字节数		描　述
S7-400	S7-300		S7-400	S7-300	
SFB8	FB8	U_SEND	440	160	与接收方通信功能（U_RCV）执行序列无关的快速的无需确认的数据交换，例如传送操作与维护消息，对方接收到的数据可能被新的数据覆盖
SFB9	FB9	U_RCV			
SFB12	FB12	B_SEND	64K	32K	将数据块安全地传输到通信伙伴，直到通信伙伴的接收功能（B_RCV）接收完数据，数据传输才结束
SFB13	FB13	B_RCV			
SFB14	FB14	GET	400	160	程序控制读取远方 CPU 的变量，通信伙伴不需要编写通信程序
SFB15	FB15	PUT			程序控制写变量到远方 CPU，通信伙伴不需要编写通信程序



有 S7-300 集成的 DP 通信接口参与时，只能进行单向 S7 通信，S7-300 集成的 DP 通信接口在通信中只能作服务器。S7-400 CPU 集成的 DP 接口在单向 S7 通信中既可以作服务器，也可以作客户机。S7-400 CPU 之间还可以进行双向 S7 通信。

网络组态工具 NetPro 有很强的防止出错的功能，它会禁止建立那些选用的硬件不支持的通信连接组态。

PLCSIM V5.4 SP3 及更高的版本支持对 S7 通信的仿真。

6.2.2　实训三十八　基于 DP 网络的单向 S7 通信

1．组态硬件

在 STEP 7 中创建一个名为“S7_DP”的项目（见随书光盘的同名例程），CPU 为 CPU 412-2DP。

选中 SIMATIC 管理器左边窗口出现的“SIMATIC 400 站点”，双击右边窗口中的“硬件”图标，打开硬件组态工具 HW Config。可以看到自动生成的机架和 4 号槽中的 CPU 模块。将电源模块和信号模块插入机架。

用鼠标双击机架中 CPU 412-2DP 下面“DP”所在的行，单击出现的 DP 属性对话框的“常规”选项卡中的“属性”按钮（见图 6-5），在出现的“属性 - PROFIBUS 接口 DP”对话框中，单击“新建”按钮，出现“属性 – 新建子网 PROFIBUS”对话框，采用默认的传输速率 1.5 Mbit/s，配置文件为“标准”。多次单击“确定”按钮，返回 HW Config，可以看到生成的名为“PROFIBUS（1）”的网络。CPU 集成的 DP 接口和 MPI 接口默认的地址均为 2，默认的工作模式为 DP 主站。单击工具栏上的按钮，编译并保存组态信息。

在 SIMATIC 管理器中生成一个 S7-300 站。在 HW Config 中，将 CPU 313C-2DP 插入机架，在自动打开的“属性 - PROFIBUS 接口”对话框的“参数”选项卡中，设置站地址为 3，选中“子网”列表中的“PROFIBUS（1）”（见图 6-22），将 CPU 313C-2DP 连接到 DP 网络上，默认的工作方式为 DP 主站。在 CPU 属性对话框的“常规”选项卡中，设置 MPI 地址为 3。将电源模块和信号模块插入机架。单击工具栏上的按钮，编译并保存组态信息。

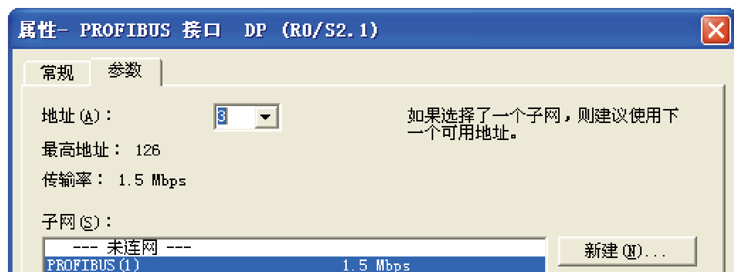


图 6-22　DP 接口属性对话框

2．组态 S7 连接

单击 SIMATIC 管理器工具栏上的按钮，打开网络组态工具 NetPro，可以看到两个站已经连接到 DP 网络上。选中 CPU 412-2DP 所在的小方框，在 NetPro 下面的窗口出现连接表（见图 6-23）。双击连接表的第 1 行，在出现的“插入新连接”对话框中（见图 6-24 的左

图），系统默认的通信伙伴为同一项目中的 CPU 313C-2DP，默认的连接类型为 S7 连接。

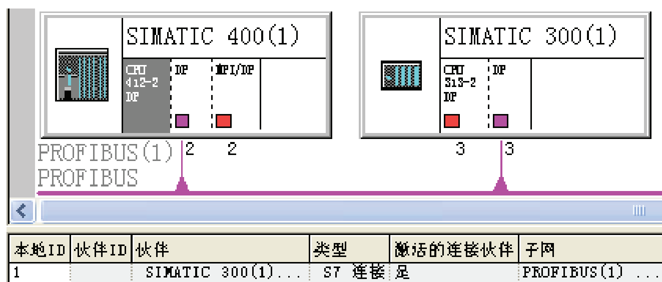


图 6-23　网络与连接的组态

单击“确定”按钮，确认默认的设置，出现 S7 连接属性对话框，因为 S7-300 集成的 DP 接口不支持双向 S7 通信，“本地连接端点”区中的“单向”复选框被自动选中，并且不能更改，默认的连接方式为“单向”。在调用通信 SFB 时，将会使用“块参数”区的“本地 ID”（本地标识符）的值。

因为是单向连接，连接表中没有通信伙伴的 ID（标识符，见图 6-23），选中 CPU 313C-2DP 所在的小方框，连接表中没有连接信息。



图 6-24　插入新连接与连接属性对话框

组态好连接后，单击工具栏上的按钮，编译并保存网络组态信息。

在单向 S7 连接中，仅需将网络组态信息下载到 S7 通信的客户机（CPU 412-2DP）。

3．生成数据块

选中 SIMATIC 管理器左边窗口中 SIMATIC 400（1）站的“块”文件夹，用鼠标右键单击右边的窗口，执行出现的快捷菜单中的命令，生成数据块 DB1。打开 DB1，生成一个名



为 ARAY，有 20 个字节元素的数组（见图 6-20）。

用同样的方法生成数据块 DB2，在数据块中创建一个名为 ARAY，有 20 个字节元素的数组。也可以用复制和修改名称的方法来创建内部结构相同的数据块。

4．S7 通信编程

在 S7 单向连接中，CPU 412-2DP 和 CPU 313C-2DP 分别作为客户机和服务器，客户机在 OB1 中调用单向通信功能块 GET 和 PUT，读、写服务器的存储区。服务器不需要调用通信功能块。本例程中的程序主要用来验证 S7 通信是否成功，没有什么工程意义。

GET、PUT 在通信请求信号 REQ 的上升沿时激活数据传输，为了实现周期性的数据传输，用时钟存储器位提供的时钟脉冲作 REQ 信号。

时钟存储器的组态方法如下：打开 HW Config，双击机架中 CPU 412-2DP 所在的行，在出现的 CPU 属性对话框的“周期/时钟存储器”选项卡中（见图 3-15），单击复选框“时钟存储器”，设置用于时钟存储器的存储器字节为 MB8。MB8 的第 1 位 M8.1 的周期为 200ms（ON 100ms，OFF 100ms）。

双击打开 CPU 412-2DP 的 OB1，用“视图”菜单中的命令设置编程语言为 STL（语句表）。执行菜单命令“视图”→“总揽”，打开程序编辑器左边的指令列表窗口。打开文件夹“\库\Standard Library\System Function Blocks”，将其中的 SFB15“PUT”和 SFB14“GET”指令拖放到程序区。

CPU 413-2DP 的 OB1 的程序段 1 中的两条语句，使 M10.0 和 M8.1 的相位相反。它们分别作系统功能块 GET 和 PUT 的 REQ（通信请求）信号，它们的上升沿互差 100ms。下面是 OB1 中的程序：

```
程序段 1：时钟脉冲信号反相
        AN          M       8.1
        =           M       10.0
程序段 2： 读取通信伙伴的数据
        CALL   "GET" , DB14                  //调用 SFB14
         REQ        :=M8.1                   //上升沿时激活数据传输，每 200ms 读取一次
         ID         :=W#16#1                 //S7 连接号
         NDR        :=M0.1                   //每次读取完成产生一个脉冲
         ERROR      :=M0.2                   //错误标志，出错时为 1
         STATUS     :=MW2                    //状态字，为 0 时表示没有警告和错误
         ADDR_1     :=P#DB1.ARAY             //要读取的通信伙伴的 1 号地址区
         ADDR_2     :=P#M 40.0 BYTE 20       //要读取的通信伙伴的 2 号地址区
         ADDR_3     :=
         ADDR_4     :=
         RD_1       :=P#DB2. ARAY            //本站存放读取的数据的 1 号地址区
         RD_2       := P#M 20.0 BYTE 20      //本站存放读取的数据的 2 号地址区
         RD_3       :=
         RD_4       :=
程序段 3：向通信伙伴的数据区写入数据
        CALL   "PUT" , DB15                  //调用 SFB15
         REQ        :=M10.0                  //上升沿时激活数据交换，每 200ms 写一次
         ID         :=W#16#1                 //S7 连接号
```

```
  DONE     :=M10.1                    //每次写操作完成产生一个脉冲
  ERROR    :=M10.2                    //错误标志，出错时为 1
  STATUS   :=MW12                     //状态字，为 0 时表示没有警告和错误
  ADDR_1   :=P#DB2. ARAY              //要写入数据的通信伙伴的 1 号地址区
  ADDR_2   :=P#M 20.0 BYTE 20         //要写入数据的通信伙伴的 2 号地址区
  ADDR_3   :=
  ADDR_4   :=
  SD_1     :=P#DB1. ARAY              //存放本站要发送的数据的 1 号地址区
  SD_2     :=P#M 40.0 BYTE 20         //存放本站要发送的数据的 2 号地址区
  SD_3     :=
  SD_4     :=
```

SFB 的在线帮助给出了 STATUS 的警告或错误代码的意义。

下面是 CPU 412-2DP 的 OB35 中的程序：

```
程序段 1：每 100ms 将 DB1.DBW0 加 1
      L      DB1.DBW     0
      +      1
      T      DB1.DBW     0
```

在 CPU 412-2DP 的初始化程序 OB100 中，调用 SFC21，将数据发送区 DB1 和 MB40~MB59 中的各个字分别预置为 16#4001 和 16#4002，将 DB2 和 MB20~MB39 中的数据接收区的各个字清零。程序中的 LW20 和 LW22 是 OB100 的局部变量字。

下面是 CPU 412-2DP 的 OB100 中的程序：

```
程序段 1：初始化存放要发送的数据的地址区
      L      W#16#4001
      T      LW      20
      CALL   "FILL"                       //调用 SFC21
       BVAL     :=LW20                    //源数据
       RET_VAL:=LW22                      //故障代码
       BLK      :=P#DB1. ARAY             //目的地址区
程序段 2：初始化存放要发送的数据的地址区
      L      W#16#4002
      T      LW      20
      CALL   "FILL"                       //调用 SFC21
       BVAL     :=LW20                    //源数据
       RET_VAL:=LW22                      //故障代码
       BLK      := P#M 40.0 BYTE 20       //目的地址区
程序段 3：将存放接收的数据的地址区清零
      L      W#16#0
      T      LW      20
      CALL   "FILL"
       BVAL     :=LW20                    //源数据
       RET_VAL:=LW22                      //故障代码
       BLK      :=P#DB2. ARAY             //目的地址区
```



```
CALL    "FILL"
 BVAL      :=LW20                    //源数据
 RET_VAL:=LW22                       //故障代码
 BLK       :=P#M 20.0 BYTE 20        //目的地址区
```

CPU 313C-2DP 和 CPU 412-2DP 的 OB100 中的程序基本上相同，其区别在于前者将数据发送区 DB1 和 MB40~MB59 中的各个字分别预置为 16#3001 和 16#3002。CPU 313C-2DP 的 OB1 中没有通信程序，OB35 每 100ms 将 DB1.DBW 加 2。

5．通信的仿真实验

图 6-25 是该项目组态和编程结束后的 SIMATIC 管理器。



图 6-25　SIMATIC 管理器

单击工具栏上的按钮，打开 PLCSIM，出现名为 S7-PLCSIM1 的窗口（见图 6-26 的左图），可以将它视为一台仿真 PLC。

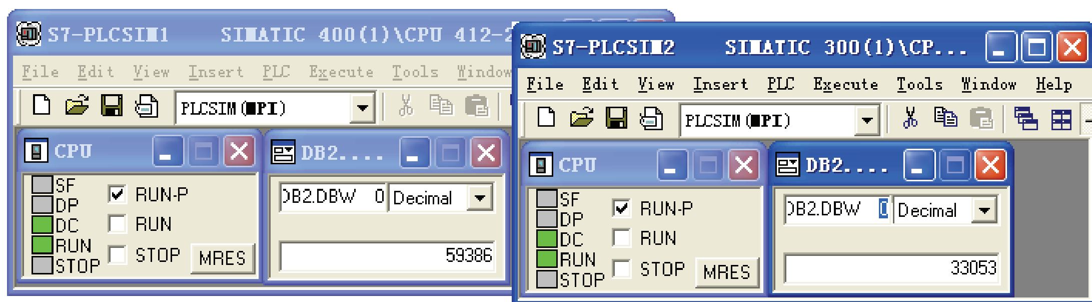


图 6-26　用 PLCSIM 仿真两台 PLC

选中 SIMATIC 管理器左边窗口 CPU 412-2DP 的“块”文件夹，单击工具栏上的按钮，下载“块”文件夹中的系统数据和程序块。下载后 S7-PLCSIM1 的标题栏出现了下载的 PLC 站的名称 SIMATIC 400(1)和 CPU 的型号 CPU 412-2DP。生成一个视图对象，将它的地址改为 DB2.DBW0。

单击 PLCSIM 工具栏上的按钮，生成一个名为 S7-PLCSIM2 的新的仿真 PLC（见图 6-26 的右图）。

选中 SIMATIC 管理器左边窗口 CPU313C-2DP 的“块”文件夹，单击工具栏上的按钮，下载“块”文件夹中的系统数据和程序块。下载后 S7-PLCSIM2 的标题栏出现了下载的

PLC 站的名称 SIMATIC 300(1)和 CPU 的型号 CPU313C-2DP。生成一个视图对象，将它的地址改为 DB2.DBW0。

在时钟脉冲 M8.1 的上升沿，CPU 412-2DP 每 200ms 读取一次 CPU 313C-2DP 的数据；在时钟脉冲 M10.0 的上升沿，每 200ms 将数据写入 CPU 313C-2DP 的数据区。

运行时通信双方的 OB35 使 DB1.DBW0 的值不断增大，然后发送给对方的 DB2.DBW0。下载后将两台仿真 PLC 切换到 RUN-P 模式，可以看到双方接收到的第一个字 DB2.DBW0 的值在不断增大。

选中 SIMATIC 管理器中的 SIMATIC 400(1)站，执行菜单命令“插入”→“S7 块”→“变量表”，生成一个变量表。双击打开生成的变量表，在地址列输入两个接收地址区的第一个字和最后一个字。

用同样的方法生成 SIMATIC 300(1)站的变量表，两个变量表中被监控的变量相同。

同时打开两个变量表，调节它们的位置和大小，保证能同时看到其中的全部变量。单击工具栏上的按钮，选中的变量表进入监控状态，“状态值”列显示的是 PLC 中变量的值，标题栏出现浅蓝色的横条。单击另一个变量表，它被选中。单击工具栏上的按钮，使它也进入监控状态。

图 6-27 是在运行时复制的通信双方的变量表。在变量表中可以看到双方接收到的 DB2.DBW0 的值在不断地增大，此外可以看到各数据接收区接收到的数据与对方 OB100 中设置的值相同。

变量 - VAT_1

VAT_1 -- @S7_DP\S...

	地址	显示格式	状态值
1	DB2.DBW 0	HEX	W#16#A02C
2	DB2.DBW 18	HEX	W#16#3001
3	MW 20	HEX	W#16#3002
4	MW 38	HEX	W#16#3002

VAT_1 -- @S7_DP\SI...

	地址	显示格式	状态值
1	DB2.DBW 0	HEX	W#16#BACB
2	DB2.DBW 18	HEX	W#16#4001
3	MW 20	HEX	W#16#4002
4	MW 38	HEX	W#16#4002

图 6-27　两块 CPU 的变量表

6.2.3　实训三十九　基于 MPI 网络的双向 S7 通信

1．硬件组态

使用 SFB BSEND/BRCV，可以进行快速的、可靠的数据传送。只有 S7-400 之间，才可以进行双向 S7 通信。本实训中通信的双方都要调用 SFB BSEND/BRCV 来发送数据和接收数据。

用 STEP 7 的“新建项目”向导创建一个名为“S7_MPI”的项目（见随书光盘中的同名例程），CPU 为 CPU 412-2DP。打开 HW Config，将电源模块和信号模块插入机架。CPU 的 MPI 地址和 DP 地址均为默认值 2，将它连接到 MPI 网络上。单击工具栏上的按钮，编



译并保存硬件组态信息。

在 SIMATIC 管理器中生成一个名为 SIMATIC 400（2）的 S7-400 站点（见图 6-28）。在 HW Config 中，将电源模块、CPU 412-2DP 和信号模块插入机架，在 CPU 属性对话框的“常规”选项卡中，设置 DP 地址和 MPI 地址均为 3，将它连接到 MPI 网络上。单击工具栏上的按钮，编译并保存组态信息。

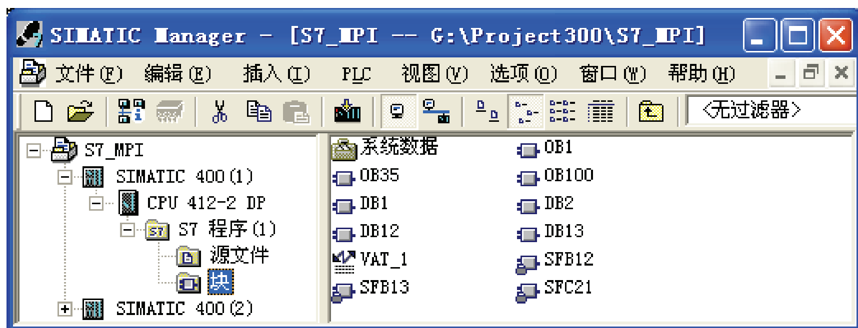


图 6-28　SIMATIC 管理器

2. 组态 S7 连接

组态好两个 S7-400 站后，单击工具栏上的按钮，打开 NetPro，看到连接到 MPI 网络上的两个站（见图 6-29）。选中“SIMATIC 400 (1)”站点的 CPU 412-2DP 所在的小方框，在下面的窗口出现连接表，双击连接表第一行的空白处，建立一个新连接。

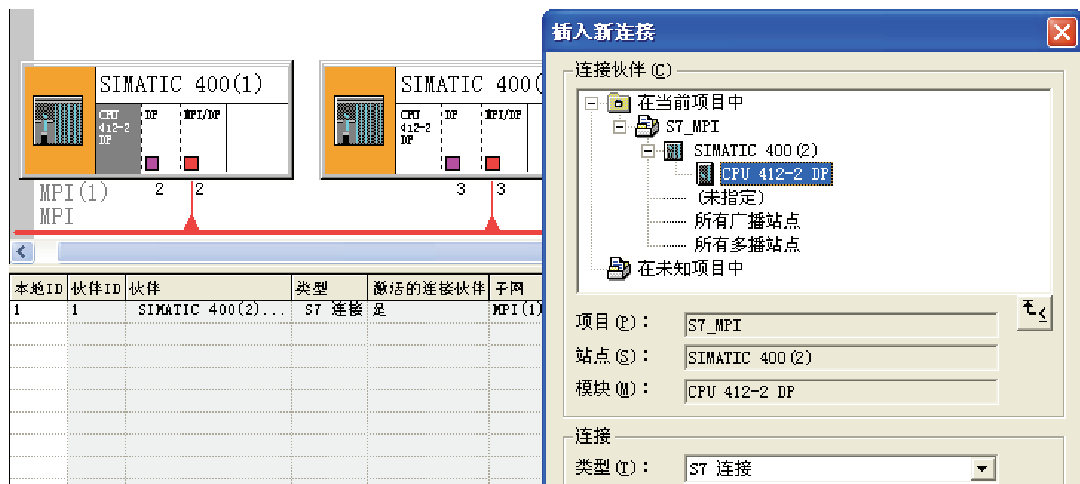


图 6-29　网络与连接的组态

在出现的“插入新连接”对话框中，系统默认的通信伙伴为站点 SIMATIC 400(2)的 CPU 412-2DP，“类型”选择框中默认的连接类型为 S7 连接。

单击“确定”按钮，出现 S7 连接属性对话框（见图 6-30 的左图）。在“本地连接端点”区，复选框“单向”被禁止选中（该复选框为灰色），因此连接是双向的，在图 6-29 的连接表中，生成了相同的“本地 ID”和“伙伴 ID”。

复选框“建立主动的连接”是默认的设置（见图 6-30 中的左图），选中该复选框时，连接表的“激活的连接伙伴”列将显示“是”。在运行时，由本地节点建立连接。反之显示“否”，由通信伙伴建立连接。

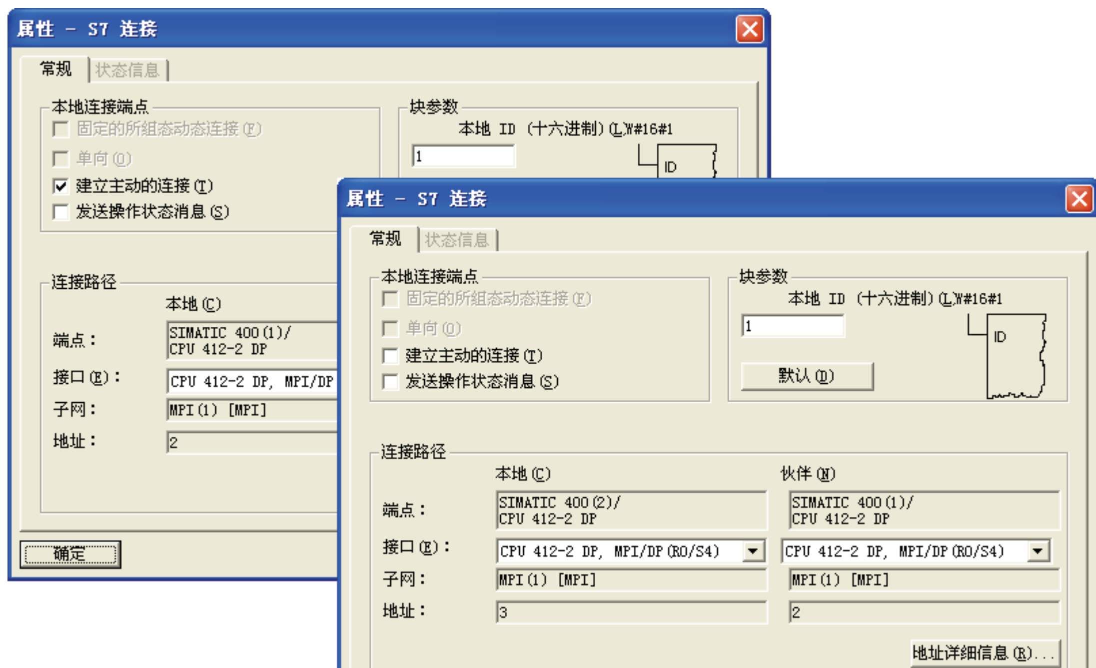


图 6-30　通信双方的 S7 连接属性对话框

选中 NetPro 中站点 SIMATIC 400(2)的 CPU 412-2DP 所在的小方框，下面的窗口是自动生成的该站点一侧的连接表（见图 6-31），双击连接表中的“S7 连接”，出现该站点一侧的连接属性对话框（见图 6-30 中的右图）。

本地ID	伙伴ID	伙伴	类型	激活的连接伙伴	子网
1	1	SIMATIC 400(1)...	S7 连接	否	MPI(1) [MPI]

图 6-31　站点 SIMATIC 400（2）一侧的 S7 连接表

组态好连接后，单击工具栏上的按钮，网络组态信息被编译和保存在系统数据中。

对于双向通信，应将通信双方的连接表信息分别下载到各自的 CPU。编译成功后，也可以通过 SIMATIC 的“块”文件夹中的“系统数据”下载硬件和连接的组态信息。

3．通信程序设计

双方的通信程序基本上相同。首先生成 DB1 和 DB2，在数据块中生成有 200 个字节元素的数组 ARAY。然后生成 OB35 和 OB100。

SFB BSEND/BRCV 的输入参数 ID 为连接的标识符，R_ID 用于区分同一连接中不同的 SFB 调用。对于同一数据包，发送方与接收方的 R_ID 应相同。站点 SIMATIC 400(1)发送和接收的数据包的 R_ID 分别为 1 和 2，站点 SIMATIC 400(2)发送和接收的数据包的 R_ID 分别为 2 和 1。



为了实现周期性的数据传输，用周期为 100ms 的时钟存储器位 M8.0 为 BSEND 提供发送请求信号 REQ。在组态硬件时，在 CPU 属性对话框的“周期/时钟存储器”选项卡设置 MB8 为时钟存储器字节（见图 3-15）。

打开 SIMATIC 400(1)的 OB1，用“视图”菜单中的命令设置编程语言为 STL（语句表）。执行菜单命令“视图”→“总揽”，打开程序编辑器左边的指令列表窗口。打开文件夹“\库\Standard Library\System Function Blocks”，将其中的 SFB12“BSEND”和 SFB13“BRCV”指令拖放到程序区。下面是站点 SIMATIC 400(1)的 OB1 中的程序。

```
程序段 1：  发送数据
      CALL   "BSEND" , DB12                    //调用 SFB12
       REQ      :=M8.0                         //上升沿时激活数据交换，周期为 100ms
       R        :=M10.1                        //上升沿时中断正在进行的数据交换
       ID       :=W#16#1                       //S7 连接号
       R_ID     :=DW#16#1                      //发送与接收请求号
       DONE     :=M10.2                        //任务被正确执行时为 1
       ERROR    :=M10.3                        //错误标志位，为 1 时出错
       STATUS   :=MW12                         //状态字
       SD_1     :=P#DB1.ARAY                   //存放要发送的数据的地址区
       LEN      :=MW14                         //要发送的数据字节数
 程序段 2：  接收数据
      CALL   "BRCV" , DB13                     //调用 SFB13
       EN_R     :=TRUE                         //接收请求，为 1 时允许接收
       ID       :=W#16#1                       //S7 连接号
       R_ID     :=DW#16#2                      //发送与接收请求号
       NDR      :=M0.1                         //任务被正确执行时为 1
       ERROR    :=M0.2                         //错误标志位，通信出错时为 1
       STATUS   :=MW2                          //状态字
       RD_1     :=P#DB2. ARAY                  //存放接收的数据的地址区
       LEN      :=MW4                          //已接收的数据字节数
```

BSEND 的输入参数 LEN 是要发送的数据的字节数，数据类型为 WORD（无符号的字）。因为不能使用常数，设置 LEN 的实参为 MW14，在双方的初始化程序 OB100 中，用下面两条语句预置它的初始值为 200：

```
L     200
T     MW     14                                //预置要发送的字节数
```

在初始化程序 OB100 中，调用 SFC21，将双方的数据发送区 DB1 的各个字分别预置为 16#4001 和 16#4002，将 DB2 中的数据接收区的各个字清零。

通信双方的 OB35 中的程序每 100ms 将发送的第一个字 DB1.DBW0 加 1。

4．通信的仿真实验

用实训三十八所述的方法，打开 PLCSIM，生成两台仿真 PLC，其中各生成一个视图对象，将它的地址改为 DB2.DBW0（见图 6-32）。分别将两个站的系统数据和程序块下载到各自的仿真 PLC。将两台仿真 PLC 切换到 RUN-P 模式。



图 6-32　用 PLCSIM 仿真两台 PLC

在时钟脉冲 M8.0 的上升沿，通信的双方每 100ms 发送 200B 数据。运行时可以看到双方接收到的第一个字节 DB2.DBW0 的值在不断增大。

图 6-33 是在运行时复制的通信双方的变量表。可以看到双方接收到的 DB2.DBW0 在不断地变化，还可以看到数据接收区的第二个字 DBW2 和最后一个字 DBW198 的值与发送方预置的相同。



图 6-33　两块 CPU 的变量表

6.2.4　实训四十　上传项目文件

STEP 7 的项目文件是控制系统的监控、调试、故障诊断和改进的基础。如果下载到硬件 CPU 的项目文件没有加密，建立起 CPU 与 STEP 7 的通信连接后，可以将项目文件上传到计算机，用 STEP 7 的项目将硬件、网络的组态信息和用户程序保存起来。

本实训通过仿真操作来学习项目文件的上传方法。

1．上传单主站项目的仿真实验

1）首先打开随书光盘中的项目“ET 200”，然后打开 PLCSIM。选中 SIMATIC 管理器左边窗口中的“块”，单击工具栏上的按钮，将组态信息和用户程序下载到仿真 PLC。

2）执行菜单命令“文件”→“新建”，在 STEP 7 中生成一个空的项目。

3）执行菜单命令“PLC”→“将站点上传到 PG”，单击出现的“选择节点地址”对话框中的“视图”按钮（见图 6-34），将会显示 MPI 网络中可访问的站点，其中的模块型号是仿真 PLC 的型号。“视图”按钮上的字符变为“更新”。将插槽由 0 改为 2，单击“确定”按钮，出现“上传给 PG”对话框，开始上传站点的系统数据和块。上传的站点的信息保存在打开的项目中，该项目原来的内容将被覆盖。



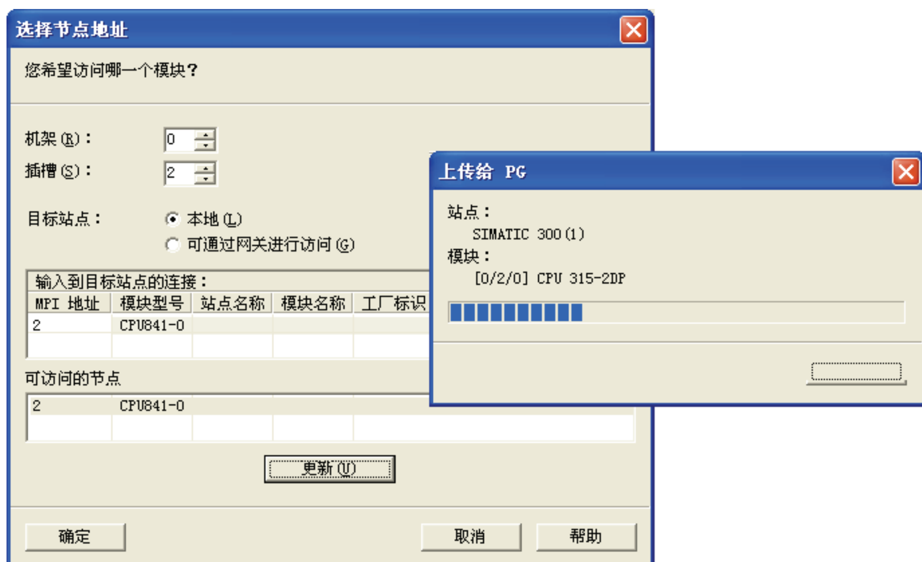


图 6-34　上传站点

4）打开上传的 300 站点，可以看到“块”文件夹中的系统数据和程序块。打开硬件组态工具 HW Config，可以看到机架中的模块、DP 网络和网络上的 ET 200M 从站。

因为没有下载，上传的项目文件没有符号表和用户程序中的注释，这样的程序的可读性很差，但是可以用于下一章将要介绍的故障诊断。

2．上传多主站项目的仿真实验

1）首先打开随书光盘中的项目“S7_DP”，该项目有两个站点，CPU 分别为 CPU 412-2DP 和 CPU 313C-2DP。打开 PLCSIM，出现名为 S7-PLCSIM1 的窗口。

2）选中 SIMATIC 管理器左边窗口 400 站点的“块”文件夹，单击工具栏上的按钮，下载系统数据和程序块。

3）单击 PLCSIM 工具栏上的按钮，生成一个名为 S7-PLCSIM2 的新的仿真 PLC。

4）选中 SIMATIC 管理器左边窗口 300 站点的“块”文件夹，单击工具栏上的按钮，下载系统数据和程序块。

5）执行菜单命令“文件”→“新建”，在 STEP 7 中生成一个空的项目。

6）执行菜单命令“PLC”→“将站点上传到 PG”，单击出现的“选择节点地址”对话框中的“视图”按钮，“可访问的节点”列表中显示出两个可访问的站点（见图 6-35 的左图）。选中其中的 400 站点（MPI 网络中的 2 号站），将插槽由 0 改为 4（应与实际的硬件组态一致），单击“确定”按钮，出现“上传给 PG”对话框，开始上传站点的系统数据和程序块。上传结束后，可以看到项目中的 400 站点。

7）再次执行菜单命令“PLC”→“将站点上传到 PG”，单击出现的“选择节点地址”对话框中的“视图”按钮，选中“可访问的节点”列表中的 3 号站（见图 6-35 中的右图），将插槽由 0 改为 2。单击“确定”按钮，出现“上传给 PG”对话框，开始上传站点的系统数据和程序块。上传结束后，可以看到项目中的 300 站点。

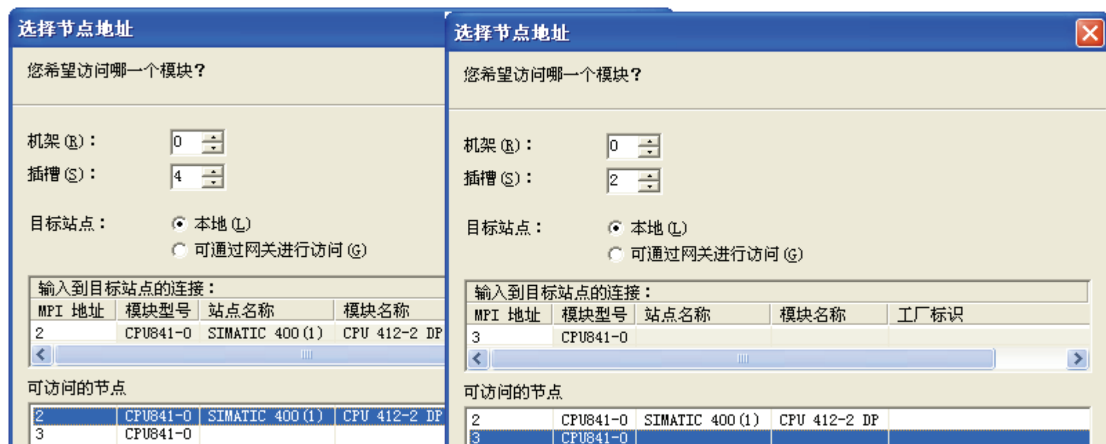


图 6-35　上传两个站点

8）打开上传的 300 站点和 400 站点，可以看到“块”文件夹中的系统数据和程序块。

9）单击工具栏上的按钮，打开网络组态工具 NetPro，可以看到 DP 网络上的两个站点。选中 400 站点中的 CPU，在下面的连接表可以看到下载的 S7 连接。

3．上传程序的仿真练习

打开随书光盘中的项目“S7_MPI”，将项目中两个 400 站点下载到仿真 PLC。

生成一个新的项目，将下载的两个站点上传到项目。检查上传的项目的程序块和硬件组态、网络组态是否正确。

6.3　练习题

1．SIMATIC NET 分为哪几层？每一层各有什么作用？各层使用什么网络？
2．PROFIBUS 有哪 3 种通信协议？
3．使用 ET 200M 有什么好处？
4．STEP 7 怎样分配 DP 网络中的 I/O 地址？
5．GSD 文件有什么作用？怎样安装 GSD 文件？
6．DP 智能从站有什么特点？怎样组态智能从站？
7．怎样实现 DP 主站与智能从站之间的一致性数据传输？
8．S7 通信可以用于哪些网络？怎样实现 S7 通信？
9．简述客户机和服务器在 S7 通信中的作用。
10．什么 CPU 集成的通信接口可以用做 S7 通信的客户机？
11．S7 单向通信和双向通信分别使用什么 SFB？
12．怎样实现两台 PLC 之间的 S7 通信的仿真？需要使用什么版本的 PLCSIM？

第 7 章

故障诊断

现代网络控制系统的站点越来越多，网络越来越复杂，对网络控制系统的故障诊断的要求越来越高。S7-300/400 提供了多种多样的故障诊断和故障显示的方法，供用户检查和定位网络控制系统的故障。本章主要介绍一些简便实用的故障诊断方法。

7.1　DP 从站与扩展机架的故障诊断

7.1.1　与 DP 通信有关的中断组织块

CPU 在识别到一个故障或编程错误时，将会调用对应的中断组织块（OB），应生成这些 OB，通过在 OB 中编写的程序对故障进行处理。下面介绍与通信故障有关的几个主要的中断组织块。

1．DP 从站产生的诊断中断（OB82）

具有诊断功能并启用了诊断中断的模块检测到错误，以及错误消失时，产生诊断中断，CPU 的操作系统将自动调用处理诊断中断的组织块 OB82。

模块通过产生诊断中断来报告事件，例如信号模块导线断开、I/O 通道的短路或过载、模拟量模块的电源故障等。OB82 的启动信息（20B 局部变量）提供产生故障的模块是输入模块或输出模块、模块的地址和故障的种类。当 DP 主站的 CPU 从 RUN 模式切换到 STOP 模式时，智能从站将调用 OB82。

通过在 OB82 中调用故障诊断的程序块（例如 SFC13），可以判断在哪个通道发生了什么样的故障。

2．优先级错误中断（OB85）

以下情况将会触发优先级错误中断：

1）产生了一个中断事件，但是没有将对应的 OB 块下载到 CPU（不包括 OB81）。

2）操作系统访问模块时出错。

3）由于通信或组态的原因，模块不存在或有故障，刷新过程映像表时 I/O 访问出错。出现故障的 DP 从站的输入/输出值装入 S7 CPU 的过程映像表时，就可能出现上述情况。

访问出错的输入字节被复位和保持为“0”，直到故障消失。

在硬件组态时双击机架中的 CPU，打开 CPU 的属性对话框。可以用“周期/时钟存储器”选项卡中的选择框设置 I/O 访问错误时调用 OB85 的方式（见图 7-1）。

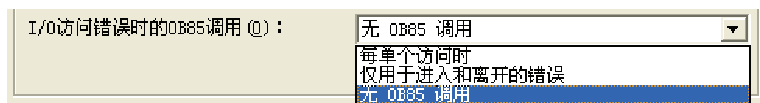


图 7-1　设置调用 OB85 的方式

S7-300 CPU 默认的选项是“无 OB85 调用”，在发生 I/O 访问错误时不调用 OB85，也不会在诊断缓冲区中生成条目。如果 S7-300 采用默认的设置，不用生成和下载 OB85。

S7-400 CPU 默认的选项是“每单个访问时”，在满足条件时，每个扫描循环周期都要调用一次 OB85 和在诊断缓冲区中生成一个条目，这样会增加扫描周期。建议选用“仅用于进入和离开的错误”，该选项只是在错误刚发生和刚消失时分别调用一次 OB85。

S7-400 的 DP 从站电源断电时，CPU 不能访问该从站，刷新过程映像表时 I/O 访问出错，如果采用默认的设置，将在每个扫描循环周期调用一次 OB85。

3．机架故障或分布式 I/O 的站故障中断（OB86）

如果扩展机架、DP 主站系统或分布式 I/O（DP 从站或 PROFINET IO 设备）由于掉电、总线导线断开、I/O 系统的故障，或者某些其他原因引起的故障，CPU 的操作系统将调用组织块 OB86。故障出现和消失时将分别调用一次 OB86。可以在 OB86 中编程保存 OB86 的启动信息，确定是哪个机架或分布式设备有故障或通信中断。

4．I/O 访问错误中断（OB122）

同步错误是与执行用户程序有关的错误，程序中如果有不正确的地址区、错误的编号或错误的地址，都会出现同步错误，操作系统将调用同步错误 OB。OB122 用于处理模块访问错误。

同步错误 OB 的优先级与检测到出错的块的优先级一致。因此 OB121 和 OB122 可以访问中断发生时的累加器和其他寄存器的内容。用户程序可以用它们来处理错误。

S7-300/400 的外设输入区/外设输出区（PI/PQ 区）用于直接读写 I/O 模块。I/Q 区是 CPU 内的过程映像存储区。

CPU 如果用 PI/PQ 地址访问有故障的 I/O 模块、不存在的或有故障的 DP 从站（例如从站断电），CPU 的操作系统将在每个扫描周期调用一次 OB122。

5．故障处理中断组织块的作用

DP 从站出现故障时，如果 S7-400 没有生成和下载 OB82、OB85、OB86 和 OB122，S7-300 没有生成和下载 OB82、OB86 和 OB122，CPU 将切换到 STOP 状态。

为了防止网络通信的故障造成 CPU 和整个 PROFIBUS 主站系统停机，作为一个常规的措施，至少要生成和下载上述组织块。即使没有在这些 OB 中编写任何程序，在 DP 从站出现上述故障时，CPU 也不会进入 STOP 模式。

需要注意的是，生成上述 OB 后，CPU 虽然不再进入 STOP 模式，但是可能不易察觉这些危险状态，它们会被忽视。为了解决这一问题，在故障 OB 中，应编写记录、处理和显示故障的程序，例如记录中断的次数，保存 OB 的局部变量，调用读取诊断数据的 SFC13 等。以便在出现故障时，迅速地查明故障的原因和采取相应的措施。

通过中断组织块的局部变量提供的信息，可以获得故障的原因、出现故障的模块地址、模块的类型（输入模块或输出模块）、是故障出现还是故障消失等信息。CPU 的模块信息对话框中的诊断缓冲区保留着 CPU 请求调用组织块的信息。



7.1.2　实训四十一　DP 从站的故障诊断

本节通过 PLCSIM 的仿真实验，来了解 PROFIBUS 网络故障的主要诊断方法。

1．用仿真软件模拟 DP 从站的故障

生成一个名为“从站诊断”的项目，其硬件结构与项目“ET 200”相同（见图 6-4），DP 主站为 CPU 315-2DP，3 号和 4 号 DP 从站为 ET 200M 和 ET 200B。

单击工具栏上的按钮，对组态信息进行编译。打开 PLCSIM，下载系统数据和 OB1。将仿真 CPU 切换到 RUN-P 模式。

执行 PLCSIM 的菜单命令“Execute”（执行）→“Trigger Error OB”（触发错误 OB）→“Rack Failure (OB86)”（机架故障 OB86），打开“Rack failure OB（86)”对话框（见图 7-2 的右图）。在“DP Failure”（DP 故障）选项卡，组态的 3 号站和 4 号站为绿色。如果有多个 DP 网络，可以用“Subnet”下拉式列表设置网络号。选中 4 号站（ET 200B），小方框内出现“×”。用单选框选中“Station failure”（站故障），单击“Apply”（应用）按钮。4 号站对应的小方框中的“×”消失，小方框变为红色，表示 4 号站出现故障。

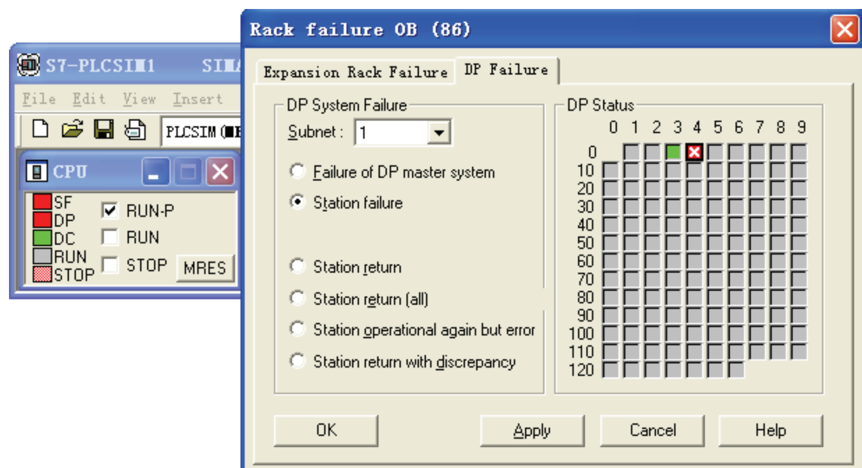


图 7-2　模拟 DP 从站故障

单击“OK”按钮，将执行与单击“Apply”按钮同样的操作，同时关闭对话框。

出现 DP 站故障时，CPU 视图对象上的红色 SF（系统故障）LED 亮，DP（总线故障）LED 闪烁。CPU 请求调用 OB86，因为没有生成和下载 OB86，CPU 自动切换到 STOP 模式，RUN LED 熄灭，STOP LED 亮。

在做硬件实验时，如果关闭 DP 从站的电源，或断开从站的通信连接，可以观察到相同的现象。

选中图 7-2 有故障的红色的 4 号站，小方框内出现“×”。用单选框选中“Station return”（站返回），单击“Apply”（应用）按钮。4 号站对应的小方框中的“×”消失，小方框变为绿色，表示 4 号站故障消失。CPU 视图对象中的 SF 和 DP LED 熄灭。

用单选框选中“Failure of DP master system”（DP 主站系统故障），模拟 DP 网络的故障。单击“Apply”按钮，网络上所有的站对应的小方框同时变为红色。用单选框选中“Station return (all)”（所有站返回），单击“Apply”按钮，网络上所有的站对应的小方框同

时变为绿色，网络故障消失。

2. 用快速视图诊断故障

4 号站有故障时，选中 SIMATIC 管理器左边窗口的 SIMATIC 300 站点，执行菜单命令“PLC”→“诊断/设置”→“硬件诊断”，打开“硬件诊断 - 快速查看”对话框（见图 7-3），该对话框又称为“快速视图”。

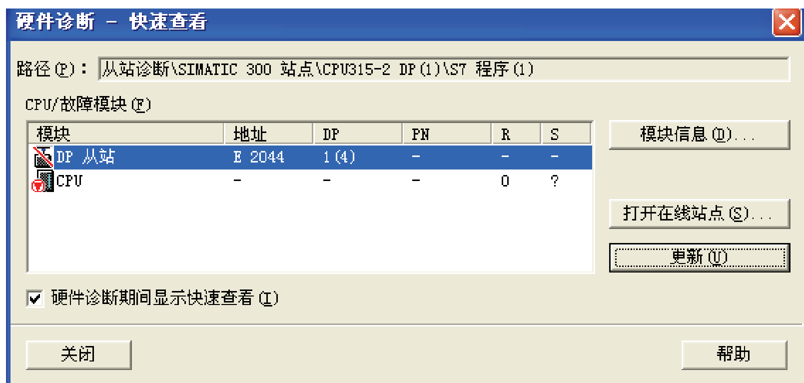


图 7-3 快速视图

“CPU/故障模块”列表给出了在线连接的 CPU 和有故障的 DP 从站的诊断符号和参数，例如模块的类型，机架号（R）和插槽号（S）。DP 列中的“1（4）”表示编号为 1 的 PROFIBUS 主站系统中的 4 号从站。PN 是 PROFINET IO 系统的编号和设备号。图中的“E”是德语 Eingang（输入）的缩写。E2044（即 IB2044）是 4 号从站的诊断地址。

图 7-3 的 CPU 上的符号表示处于 STOP 模式。DP 从站上的红色斜线表示“预设定（下载的）组态和实际组态不一样，已组态的模块不存在或已断电，通信中断，或插入了与组态的模块不同类型的模块。”

单击“帮助”按钮，或按计算机的〈F1〉键，打开快速视图的在线帮助。单击其中绿色的“诊断符号”，可以查看模块状态符号的意义。

3. 用模块信息诊断故障从站

选中快速视图中的 DP 从站，单击“模块信息”按钮，打开 4 号从站（ET 200B）的模块信息（见图 7-4）。

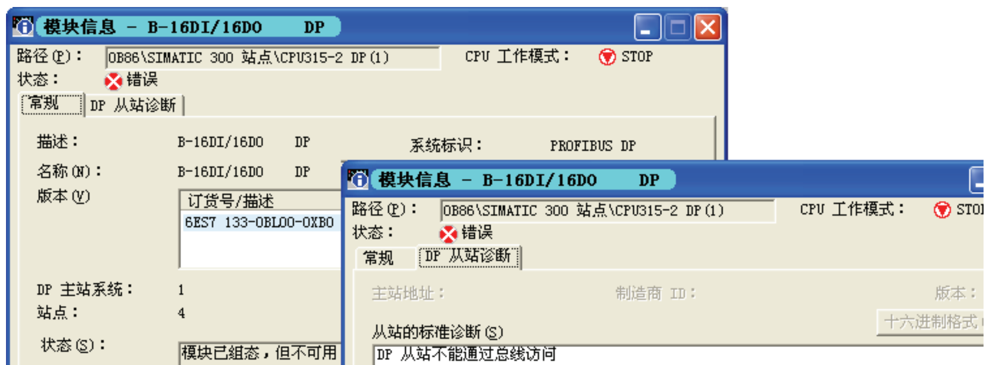


图 7-4 4 号从站的模块信息



硬件 DP 从站出现故障时，可以从模块信息获取更多的故障信息，单击“十六进制格式”按钮，可以得到该从站的十六进制的故障诊断数据。

4．用 CPU 的诊断缓冲区诊断故障

CPU 的诊断缓冲区提供了非常重要的故障诊断信息。选中快速视图中的 CPU，单击“模块信息”按钮（见图 7-3），打开 CPU 的模块信息对话框。在 SIMATIC 管理器中，选中要检查的站，执行菜单命令“PLC”→“诊断/设置”→“模块信息”，也可以打开 CPU 的模块信息对话框。

“诊断缓冲区”选项卡提供了故障和事件信息（见图 7-5），显示发生的事件的一览表和选中的事件的详细信息，可以找到使 CPU 进入 STOP 模式的原因。诊断事件按照它们发生的先后次序存储在诊断缓冲区中。CPU 进入 STOP 模式时，诊断缓冲区的内容仍然保留。

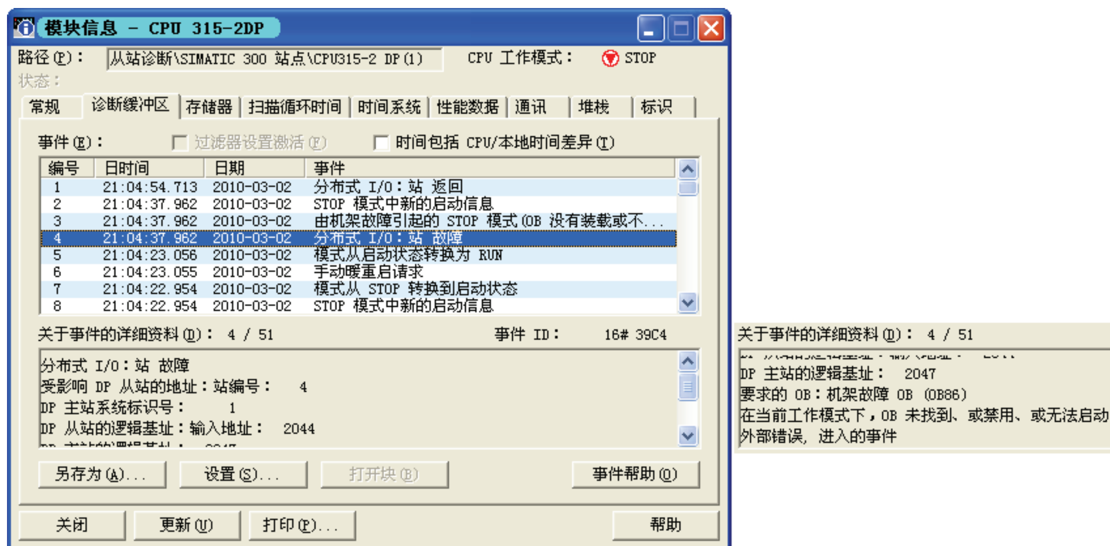


图 7-5　诊断缓冲区

图 7-5 的“事件”列表的 4 号事件“分布式 I/O：站故障”表示 4 号从站出现故障，“事件”列表第一行的“分布式 I/O：站返回”表示故障消失。

选中“事件”列表中某一行的事件，下面灰色背景的“关于事件的详细资料”窗口将显示所选事件的详细信息。图 7-5 右边的小图是 4 号事件“关于事件的详细资料”的下半部分。

选中图 7-5 的“事件”列表的 1 号事件“分布式 I/O：站返回”，故障的详细信息与 4 号事件的基本上相同。最后一行是“外部错误，离开的事件”（事件消失）。

选中诊断缓冲区中的某个事件，单击“事件帮助”按钮，将获得进一步的帮助信息。

5．OB86 的程序设计

出现 DP 站故障时，CPU 将会自动调用 OB86。在 SIMATIC 管理器中生成 OB86，不需要编写任何程序，将它下载到仿真 CPU 后，用上述方法模拟 DP 从站的故障，CPU 就不会切换到 STOP 模式。

在 OB86 中编写下面的程序，用 MW10 记录 CPU 调用 OB86 的次数。OB86 的 20B 局部变量有丰富的故障信息。生成数据块 DB86，在 DB86 中生成 20B 的数组 ARY。调用 SFC20“BLKMOV”来保存 OB86 的局部变量。

```
程序段 1：MW10 加 1
      L      MW     10
      +      1
      T      MW     10
程序段 2：保存 OB86 的局部变量
      CALL   "BLKMOV"                         //调用 SFC20
       SRCBLK  :=P#L 0.0 BYTE 20
       RET_VAL:=MW54
       DSTBLK  :=DB86.ARY
```

在 SIMATIC 管理器（见图 7-6）生成一个变量表 VAT_1。将程序块下载到仿真 PLC，打开变量表（见图 7-7），单击工具栏上的按钮，启动监控功能。可以看到在 4 号从站出现故障和故障消失时，CPU 分别调用了一次 OB86，每次调用时 MW10 的值加 1。



图 7-6　SIMATIC 管理器

4 号站有故障时，打开 DB86。单击工具栏上的按钮，启动监控功能。图 7-8 是 DB86 保存的 OB86 的 20B 局部变量。

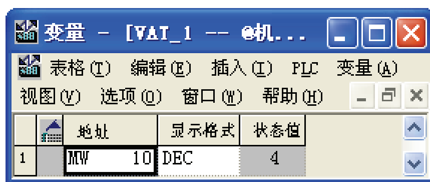


图 7-7　变量表

地址	名称	类型	初始值	实际值
0.0	ARY[1]	DWORD	DW#16#0	DW#16#39C41A56
4.0	ARY[2]	DWORD	DW#16#0	DW#16#C05407FF
8.0	ARY[3]	DWORD	DW#16#0	DW#16#07FC0104
12.0	ARY[4]	DWORD	DW#16#0	DW#16#10030112
16.0	ARY[5]	DWORD	DW#16#0	DW#16#40365642

图 7-8　OB86 中的局部变量

6. OB86 的局部变量分析

选中 SIMATIC 管理器中的 OB86，按计算机的〈F1〉键，打开 OB86 的在线帮助，可以分析图 7-8 中 OB86 各局部变量的意义：

1）DB86 的 DBB0（OB86_EV_CLASS）为 B#16#39 表示故障刚出现，为 B#16#38 表示故障刚消失。

2）下面是与 DP 通信有关的故障代码 OB86_FLT_ID（DBB1）的值的意义：

- B#16#C3：分布式 I/O 设备的 DP 主站系统故障。



- B#16#C4：DP 站故障。
- B#16#C5：DP 站内部的故障。

3）DBB2 的中断优先级（OB86_PRIORITY）为 16#1A（26）。

4）DBB3 的 OB 编号（OB86_OB_NUMBR）为 16#56（86）。

5）DBW4 保留未用。

6）DBW6 的#07FF（2047）是 DP 主站的 DP 接口的诊断地址（OB86_MDL_ADDR）。可以在 DP 接口属性对话框的“地址”选项卡找到它。

7）DBD8（OB86_RACKS_FLTD）的数据类型为 32 个位元素的数组（Array），为了方便编程，可以将它的数据类型更改为 DWORD（双字）。如果故障代码为 B#16#C4（DP 站故障），DBW8 中的 16#07FC（2044）是故障从站的诊断地址，与硬件组态中 4 号站的对应参数相同。DBW10 中的 16#0104 表示 DP 主站系统的编号为 1，从站的站地址为 4。

8）DBD12 和 DBD16（OB86_DATE_TIME）是调用 OB 的日期和时间。16#10030112 和 16#40365642 表示事件发生在 2010 年 3 月 1 日 12 点 40 分 36 秒 564 毫秒，星期一。

4 号从站故障消失时，CPU 又调用一次 OB86，MW10 加 1。OB86 的局部变量的前 12B 与从站有故障时基本上相同，其区别仅在于第一个字节为 16#38，表示事件离开（消失）。

7．DP 从站故障诊断的仿真练习

打开项目“从站诊断”后，打开 PLCSIM，将系统数据和程序块下载到仿真 CPU。将仿真 CPU 切换到 RUN-P 模式。

依次完成下列的操作：

1）用 PLCSIM 产生 3 号从站 ET 200M 的站故障。

2）观察 CPU 视图对象上 LED 的变化。

3）打开快速视图，判断 CPU 和故障从站的状态。

4）查看 3 号从站的模块信息。

5）打开 CPU 的模块信息的“诊断缓冲区”选项卡，查看故障的事件信息。

6）打开 DB86，切换到在线监控状态，可以看到 OB86 的 20B 局部变量。

7）选中 OB86，按计算机的〈F1〉键，打开 OB86 的在线帮助，分析 DB86 中 OB86 的局部变量的意义。

8）用变量表来监控调用 OB86 的次数。

9）用 PLCSIM 消除故障，重复上述的操作。

10）用 PLCSIM 产生和消除 DP 主站系统（即 DP 网络）的故障。

7.1.3　实训四十二　扩展机架的故障诊断

用“新建项目”菜单创建一个名为“机架故障”的项目（见随书光盘中的同名例程），CPU 为 CPU 315-2DP。打开 HW Config，将硬件目录窗口的文件夹“\SIMATIC 300\Rack-300”中的 Rail（导轨）拖放到硬件组态窗口，生成 1 号机架（见图 7-9）。打开硬件目录窗口的文件夹“\SIMATIC 300\IM-300”，将其中的接口模块“IM 360 IM S”插入 CPU 机架的 3 号槽，将“IM 361 IM R”插入 1 号机架的 3 号槽，两块接口模块之间自动生成一条连接线。将电源模块和信号模块插入两个机架。

如果只有两个机架，可以使用价格便宜的接口模块 IM 365。

单击工具栏上的按钮，对组态信息进行编译。打开 PLCSIM，选中 SIMATIC 管理器左边窗口中的“块”文件夹，单击工具栏上的按钮，下载系统数据和程序块。将仿真 CPU 切换到 RUN-P 模式。

将项目“从站诊断”所有的程序块（见图 7-6）复制到项目“机架故障”。

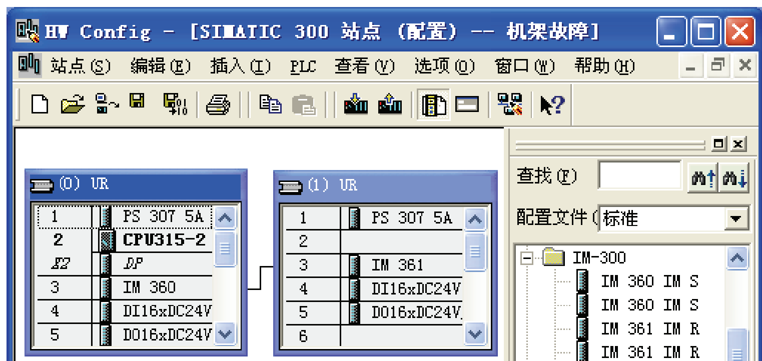


图 7-9　组态扩展机架

执行 PLCSIM 的菜单命令“Execute”（执行）→“Trigger Error OB”（触发错误 OB）→“Rack Failure (OB86)”（机架故障 OB86），打开“Rack failure OB（86）”对话框（见图 7-10）。在“Expansion Rack Failure”（扩展机架故障）选项卡，组态的 1 号机架对应的小方框为绿色。小方框内自动出现的“×”表示被选中。用单选框选中“Failure”（故障），单击“Apply”（应用）按钮，1 号机架对应的小方框变为红色，表示 1 号机架出现故障。

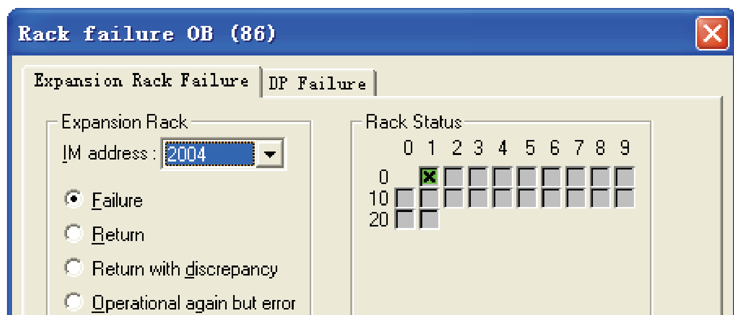


图 7-10　模拟扩展机架故障

依次完成下列的诊断操作：

1）观察 CPU 视图对象上的 LED 的变化。

2）打开 CPU 的模块信息的“诊断缓冲区”选项卡，查看扩展机架故障的事件信息。

3）打开 DB86，切换到在线监控状态，可以看到 OB86 的 20B 局部变量。

4）选中 OB86，按计算机的〈F1〉键，打开 OB86 的在线帮助，分析 DB86 中 OB86 的局部变量的意义。

5）用变量表监控调用 OB86 的次数。

6）用单选框选中图 7-10 中的“Return”（返回），1 号机架对应的小方框变为绿色。重



复上述的操作。

7.2　实训四十三　信号模块诊断功能的仿真实验

具有诊断功能的分布式 I/O 模块通过产生诊断中断来报告事件。产生诊断中断时，CPU 的操作系统将自动调用处理诊断中断的组织块 OB82。OB82 的启动信息提供了产生故障的模块的类型、模块的地址和故障的种类。

1．创建项目

将实训四十二的项目“从站诊断”另存为项目“诊断中断”（见随书光盘的同名例程）。DP 主站为 CPU 315-2DP，3 号 DP 从站为 ET 200M（图 6-4），它的 AI、AO 模块均有诊断功能。

2．组态信号模块的诊断功能

选中 3 号从站 ET 200M，双击 7 号槽的 2AO 模块，在它的属性对话框的“输出”选项卡中（见图 7-11），设置 0 号通道输出 4～20mA 的电流，1 号通道输出 0～10V 的电压。启用模块的诊断中断功能和两个通道的“组诊断”功能。

AO 模块的通道被组态为电流输出时，它的输出电阻很大，外部输出回路可以短路，如果开路则出现故障。AO 模块的通道被组态为电压输出时，它的输出电阻很小，外部输出回路可以开路，如果短路则出现故障。

按下计算机的〈F1〉键，在出现的在线帮助中，单击绿色的“诊断”，可以查看“组诊断”的帮助信息。

由帮助信息可知，组诊断可以检测组态和参数分配错误、电压输出时的接地短路、电流输出时断线和丢失负载电压 L+的故障。出现诊断事件时，CPU 将会调用诊断中断组织块 OB82，同时相应的信息会输入到 CPU 模块信息的诊断缓冲区。

双击 ET 200M 第 6 槽的 2AI 模块，在它的属性对话框的“输入”选项卡中（见图 7-12），设置测量范围为 4～20mA 的电流，启用模块的诊断中断功能、组诊断功能和断线检查功能。

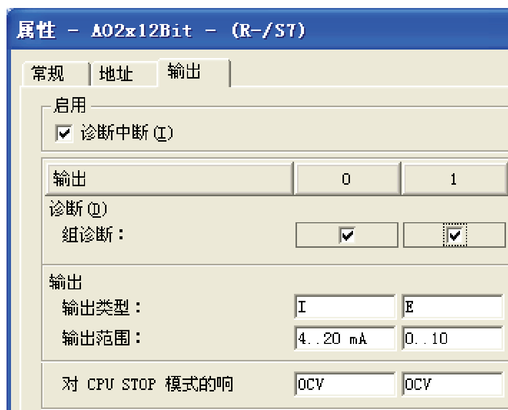


图 7-11　组态 AO 模块的诊断功能

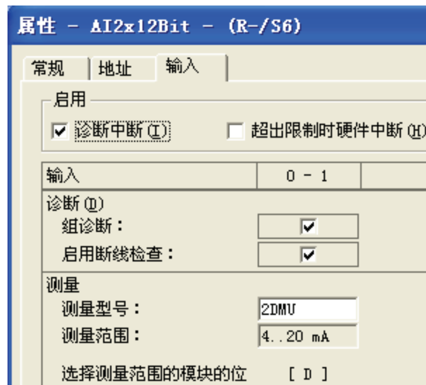


图 7-12　组态 AI 模块的诊断功能

单击工具栏上的按钮，对组态信息进行编译。

3. 编写 OB 的程序

在 SIMATIC 管理器中生成 OB82、OB85、OB86 和 OB122。

下面是 OB82 中的程序，将 MW10 加 1，用 MW10 来计调用 OB82 的次数：

```
程序段 1：MW10 加 1
      L     MW    10
      +     1
      T     MW    10
```

在 OB85、OB86 和 OB122 中编写程序，分别将 MW12~MW16 加 1。

生成数据块 DB82，在数据块中生成数组 ARY。在 OB82 中调用 SFC20“BLKMOV”，将它的局部变量保存在数组 DB82.ARY 中。

```
程序段 2：保存 OB82 的局部变量
      CALL   "BLKMOV"
       SRCBLK  :=P#L 0.0 BYTE 20
       RET_VAL:=MW50
       DSTBLK  :=DB82.ARY
```

生成一个变量表，在变量表中监控 MW10~MW16。

打开 PLCSIM，选中 SIMATIC 管理器左边窗口中的“块”文件夹，单击工具栏上的按钮，下载系统数据和程序块。将仿真 CPU 切换到 RUN-P 模式。

执行 PLCSIM 的菜单命令“Execute”（执行）→“Trigger Error OB”（触发错误 OB）→“Diagnostic interrupt (OB82)”（诊断中断 OB82），打开“Diagnostic Interrupt OB（82）”对话框（见图 7-13）。

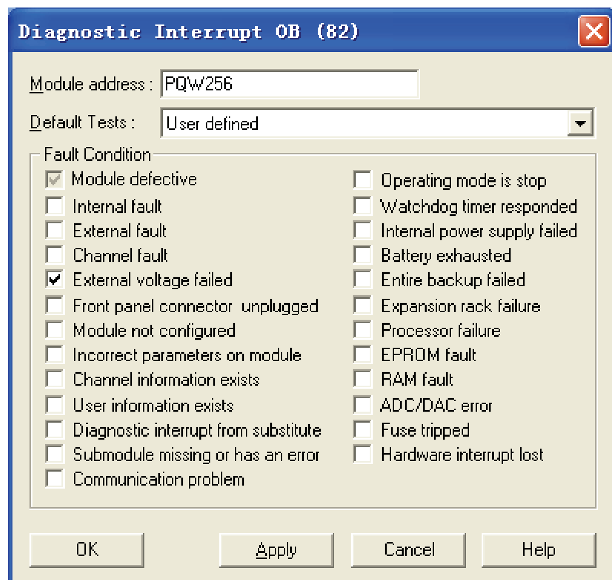


图 7-13 模拟 AO 模块的故障

在“Module address”文本框输入 AO 模块的起始地址 PQW256，用复选框选中



“External voltage failed”（外部电压故障），单击“Apply”（应用）按钮，模拟 AO 模块出现故障。

CPU 视图对象上的红色 SF（系统故障）LED 亮，因为与 DP 从站的通信正常，DP（总线故障）LED 未亮。CPU 自动调用 OB82，如果没有生成和下载 OB82，CPU 将自动切换到 STOP 模式，RUN LED 熄灭，STOP LED 亮。

单击图 7-13 中的复选框“External voltage failed”，其中的“√”消失。单击“Apply”（应用）按钮，模拟 AO 模块的诊断错误消失。CPU 视图对象上的 SF LED 熄灭。

4. 用快速视图诊断故障

AO 模块有诊断故障时，选中 SIMATIC 管理器左边窗口的 SIMATIC 300 站点，执行菜单命令“PLC”→“诊断/设置”→“硬件诊断”，打开“硬件诊断 - 快速查看”对话框（见图 7-14）。“CPU/故障模块”列表中的 3 号 DP 从站上有故障符号（红色的指示灯）。

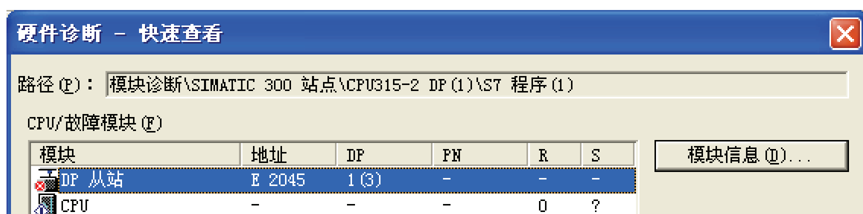


图 7-14　快速视图

选中有故障的 DP 从站，单击“模块信息”按钮，打开 3 号从站的接口模块 IM 153-1 的模块信息（见图 7-15）。

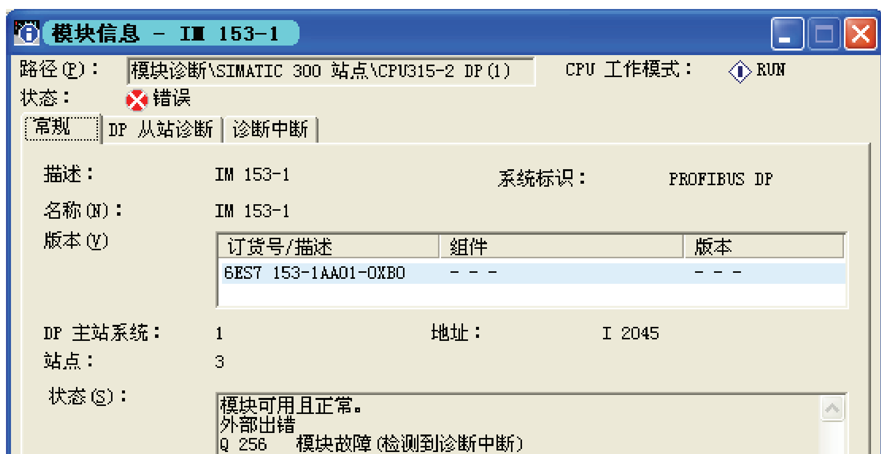


图 7-15　IM 153-1 的模块信息

5. 用 CPU 的诊断缓冲区诊断故障

选中快速视图中的 CPU，单击“模块信息”按钮，打开 CPU 的模块信息对话框。图 7-16 的“事件”列表中的 2 号事件为“模块故障存在”，右边的小图是 2 号事件的详细信息的下半部分，模块的故障是“没有外部辅助电源”。



图 7-16　CPU 的模块信息对话框

“事件”列表中的 1 号事件“模块 确定”是故障消失的信息。故障的详细信息与 2 号事件的基本上相同，最后一行是“外部错误，离开的事件”。

打开变量表，单击工具栏上的按钮，启动监控功能。可以看到在 AO 模块出现故障和故障消失时，CPU 分别调用了一次 OB82，每次调用时 MW10 的值加 1。

6．用诊断视图诊断故障模块

为了确认 3 号从站的 AO 模块的具体故障，可以使用诊断视图。诊断视图实际上就是在线的硬件组态窗口。单击快速视图中的“打开在线站点”按钮（见图 7-3），打开诊断视图（见图 7-17）。

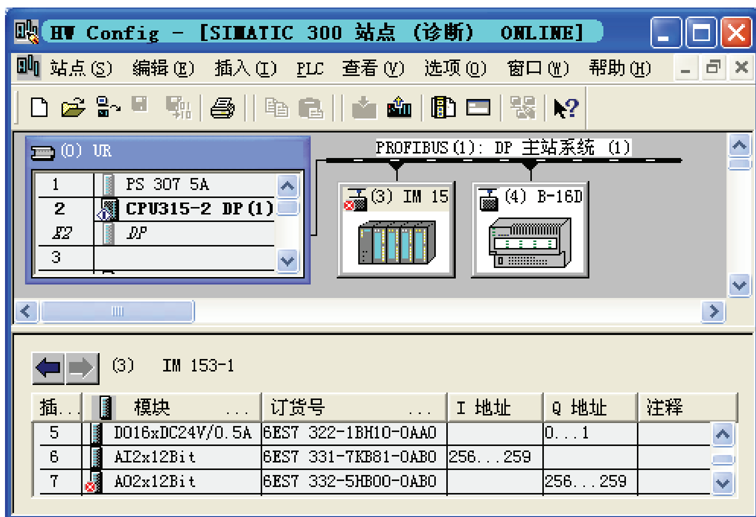


图 7-17　诊断视图



打开离线的硬件组态工具 HW Config，单击工具栏上的在线/离线切换按钮，也能打开诊断视图。

与快速视图相比，诊断视图显示整个 300 站点在线的情况，可以读取每个模块的在线状态。用这种方法可以得到那些没有故障因而没有在快速视图中显示的模块的信息。

图 7-17 中的 3 号从站 ET 200M 和该从站 7 号槽的 AO 模块上均有故障符号。双击 AO 模块，打开它的模块信息对话框（见图 7-18），可以查看该模块的详细故障信息。

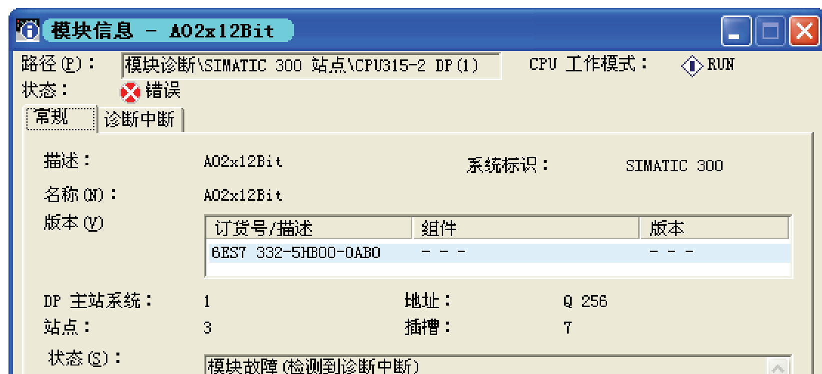


图 7-18　AO 模块的模块信息对话框

7. OB82 的局部变量中的故障信息

如果具有诊断功能的模块在组态时已经启用了诊断中断，在检测到故障产生和消失时，它将会分别向 CPU 发送一个诊断中断请求，操作系统将调用 OB82。

3 号站的 AO 模块有故障时，打开 DB82。单击工具栏上的按钮，启动监控功能。图 7-19 是 DB82 中保存的 OB82 的 20B 局部数据。

选中 SIMATIC 管理器中的 OB82，按计算机的〈F1〉键，打开 OB82 的在线帮助。由在线帮助可以获得图 7-19 中的局部变量的意义。

地址	名称	类型	初始值	实际值
0.0	ARY[1]	DWORD	DW#16#0	DW#16#39421A52
4.0	ARY[2]	DWORD	DW#16#0	DW#16#C5550100
8.0	ARY[3]	DWORD	DW#16#0	DW#16#11050000
12.0	ARY[4]	DWORD	DW#16#0	DW#16#10030209
16.0	ARY[5]	DWORD	DW#16#0	DW#16#51581383

图 7-19　OB82 的局部变量

DBB0 的 16#39 表示进入事件（事件发生）。

DBB1 的 16#42 是错误代码。

DBB2 的中断优先级为 16#1A（26）。

DBB3 的 OB 编号为 16#52（82）。

DBB4 保留未用。

DBB5 的 16#55 表示故障模块为输出模块。

DBW6 为 16#0100（256），是出现故障的 AO 模块的逻辑基地址。

DBD8 为故障模块的诊断数据，其中的 DBW8 为 W#16#1105 或 2#0001 0001 0000 0101，由在线帮助可知其含义如下：模块发生故障，外部电压故障，2#0101 是模拟量输出模块的代码。

故障消失时，OB82 的局部变量与故障出现时的基本上相同，其区别在于 DBB0 为 16#38，表示事件消失（离开事件）。此外 DBW8 由 B#16#1105 变为 B#16#0005，表示故障消失。

8．AO 模块故障诊断的硬件实验

OB82 的局部变量不能提供信号模块所有的诊断信息，例如不能提供 AO 模块的输出电路开路和短路故障的信息。为此需要在诊断视图中查看 AO 模块的模块信息，或者用 SFC13 读取故障信息。

PLCSIM 只能模拟信号模块的部分故障，不能模拟的故障必须用硬件做诊断实验。

作者做硬件实验的控制系统的硬件结构和组态与本实训的项目中的基本上相同。在 AO 模块 0 号通道的电流输出端外接一个小开关，将开关断开，模块的电流输出回路出现开路故障。CPU 315-2DP、IM 153-1 和 AO 模块上的 SF LED 亮。诊断视图中 3 号从站和 AO 模块上均有错误符号。变量表中 MW10 的值加 1，表明调用了一次 OB82。在 CPU 的模块信息对话框的诊断缓冲区的事件列表中，可以看到调用 OB82，触发中断的模块的地址和其他信息。

用小开关接通 AO 模块的电流输出电路，开路故障消失。模块上的故障 LED 熄灭，CPU 又调用一次 OB82，MW10 的值加 1。在 CPU 的模块信息对话框中，可以看到有关的信息。

用接在 1 号通道输出端的小开关将其电压输出电路短路，将会触发诊断中断，CPU 也会调用 OB82。

选中诊断视图中的 3 号从站，双击下面窗口的 AO 模块，打开 AO 模块的模块信息对话框，“常规”选项卡与图 7-18 相同。

“诊断中断”选项卡（见图 7-20）给出了模块的标准诊断信息。“指定通道的诊断”列表给出了出现故障的通道编号和具体的错误信息。选中该列表中的某个通道，单击下面的“显示”按钮，将出现帮助信息。

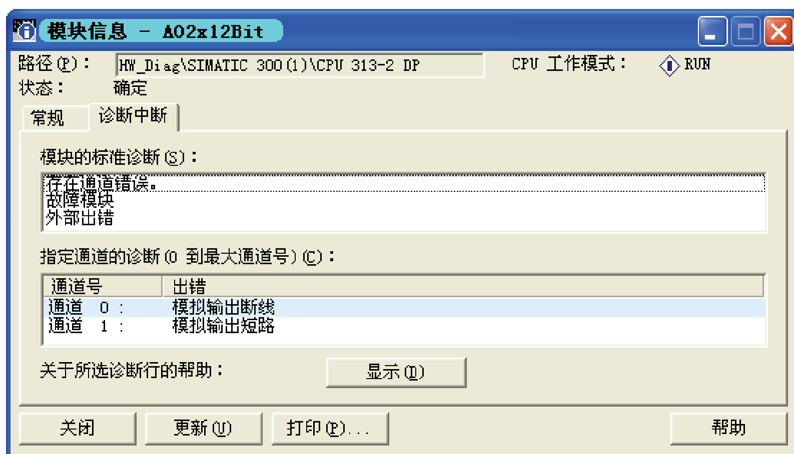


图 7-20 AO 模块的模块信息对话框



从这个例子可以看出，用本实训介绍的方法和硬件实验来诊断信号模块的故障，可以获得准确的故障信息。

9．信号模块故障诊断的仿真练习

打开项目“诊断中断”后，打开 PLCSIM，将系统数据和程序块下载到仿真 CPU。将仿真 CPU 切换到 RUN-P 模式。

依次完成下列的操作：

1）用 PLCSIM 产生 ET 200M 的 AI 模块的“Front panel connector not plugged in”（前连接器未插入）故障。

2）观察 CPU 视图对象上的 LED 的变化。

3）打开 CPU 的模块信息的“诊断缓冲区”选项卡，查看该故障的事件信息。

4）打开 DB82，切换到在线监控状态，可以看到 OB82 的 20B 局部变量。

5）选中 OB82，按计算机的〈F1〉键，打开 OB82 的在线帮助，分析 DB82 中 OB82 的局部变量的意义。

6）用变量表监控调用 OB82 的次数。

7）用 PLCSIM 消除故障，重复上述的操作。

8）用 PLCSIM 产生和消除其他诊断中断事件，重复上述的操作。

7.3　实训四十四　编程错误的诊断

1．程序结构

出现编程错误时，CPU 的操作系统将调用 OB121。用新建项目向导生成一个名为“OB121”的项目（见随书光盘中的同名例程），可以选任意型号的 CPU。

生成 FC1，在 FC1 中输入下面的程序，第一条指令访问了并不存在的 DB1 中的地址：

```
L      DB1.DBW     4
T      MW      10
```

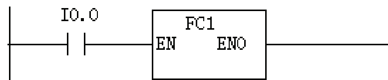


图 7-21　OB1 的程序

在 OB1 中，用 I0.0 调用 FC1（见图 7-21）。

2．仿真实验

打开 PLCSIM，选中 SIMATIC 管理器左边窗口中的“块”文件夹，单击工具栏上的按钮，下载用户程序。将仿真 CPU 切换到 RUN-P 模式。

在 PLCSIM 中生成 IB0 的视图对象。单击 I0.0 对应的小方框，将 I0.0 置为 1 状态，OB1 调用 FC1。由于 FC1 的编程错误，CPU 视图对象上的 SF LED 亮。CPU 要求调用 OB121，因为没有生成和下载 OB121，CPU 自动切换到 STOP 模式。RUN LED 熄灭，STOP LED 亮。

在 SIMATIC 管理器中执行菜单命令“PLC”→“诊断/设置”→“模块信息”，打开模块信息对话框（见图 7-22）。

选中“诊断缓冲区”选项卡的第 3 条事件“没有装载 DB”，下面的窗口是事件的详细信息：因为没有下载 DB1，程序在 FC1 中断，请求调用 OB121。图 7-22 左下角的小图是 3 号事件详细信息的下半部分。

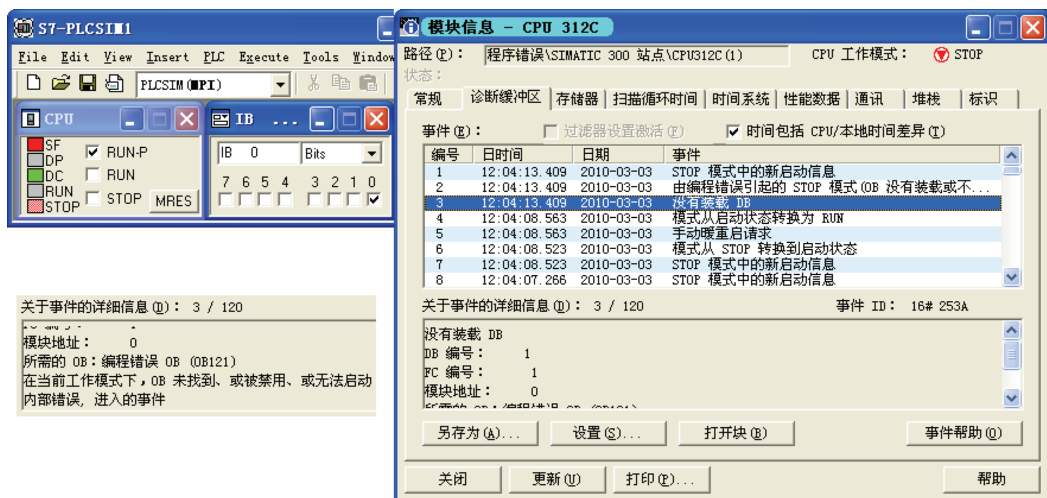


图 7-22　PLCSIM 与诊断缓冲区

单击对话框中的“打开块”按钮，将会打开出错的块 FC1，显示出错的程序段，光标在出错的第一条指令。

打开“模块信息”对话框的“堆栈”选项卡（见图 7-23），B 堆栈（块堆栈）中是与编程错误有关的块 OB1 和 FC1，表示出错时程序块的调用路径。单击该选项卡的“I 堆栈”按钮，打开中断堆栈，可以看到程序执行中断时累加器、地址寄存器和状态字的内容，在“中断点”区可以看到 FC1 的执行被中断。单击“帮助”按钮可以得到有关的帮助信息。

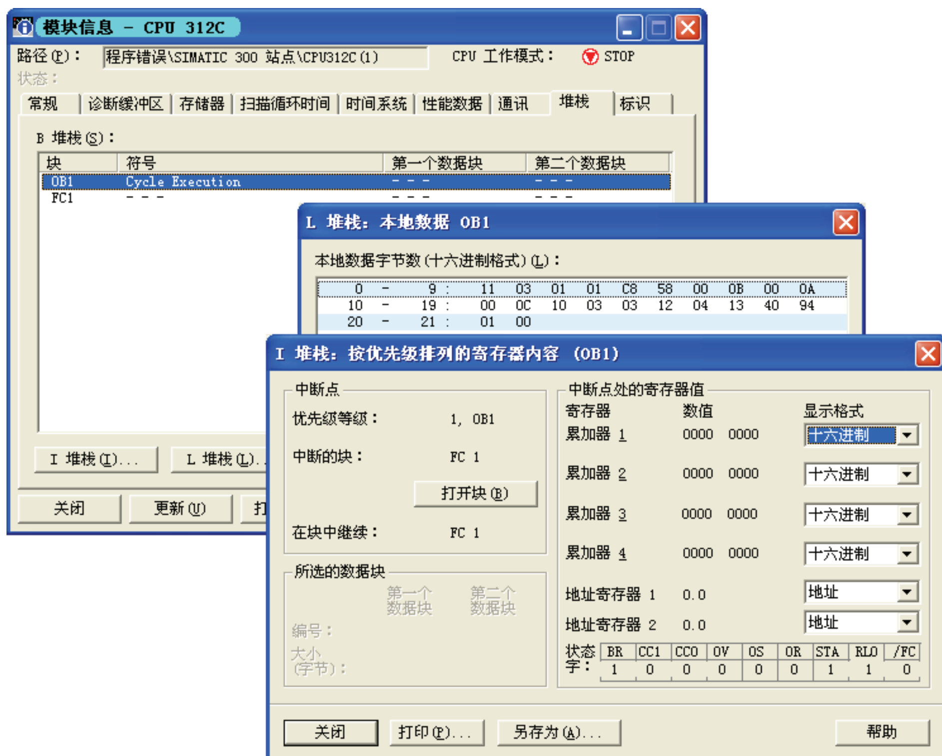


图 7-23　堆栈信息



返回 SIMATIC 管理器，生成 OB121（可以是一个空的块）和 DB121，在 DB121 中生成 20B 的数组 ARY。在 OB121 中将 MW10 加 1，调用 SFC20，用数组 DB121.ARY 保存 OB121 的局部变量。下载用户程序后切换到 RUN-P 模式。令 I0.0 为 1 状态，调用 FC1 时出现编程错误，SF LED 亮，但是 CPU 不会进入 STOP 模式。

3．OB121 的局部变量

选中 SIMATIC 管理器中的 OB121，按计算机的〈F1〉键，打开 OB121 的在线帮助。

局部变量 OB121_SW_FLT 是错误代码，OB121_FLT_REG 是错误来源，例如，产生数据类型转换错误的寄存器，不正确的地址（读/写错误），不正确的定时器/计数器和块的编号，以及不正确的存储区。OB121_RESERVED_1 给出了错误的信息（见表 7-1）。

表 7-1　OB121 的错误代码 OB121_SW_FLT

B#16#21 OB121_FLT_REG	BCD 转换错误 有关寄存器的标识符，例如，累加器 1 的标识符为 0
B#16#22 B#16#23 B#16#28 B#16#29 OB121_FLT_REG OB121_RESERVED_1	读操作时的区域长度错误 写操作时的区域长度错误 用指针读字节、字和双字时位地址不为 0 用指针写字节、字和双字时位地址不为 0 不正确的字节地址，可以从 OB121_RESERVED_1 读出数据区和访问类型 第 4～7 位为访问类型，为 0～3 分别表示访问位、字节、字和双字 第 0～3 位为存储器区，为 0～7 分别表示 I/O 区、过程映像输入表、过程映像输出表、位存储器、共享 DB、背景 DB、自己的局部数据、调用者的局部数据
B#16#24 B#16#25 OB121_FLT_REG	读操作时的范围错误 写操作时的范围错误 低字节有非法区域的标识符 (B#16#86 为自己的数据区)
B#16#26 B#16#27 OB121_FLT_REG	定时器编号错误 计数器编号错误 非法的编号
B#16#30 B#16#31 B#16#32 B#16#33 OB121_FLT_REG	对有写保护的全局 DB 的写操作 对有写保护的背景 DB 的写操作 访问共享 DB 时的 DB 编号错误 访问背景 DB 时的 DB 编号错误 非法的 DB 编号
B#16#34 B#16#35 B#16#3A B#16#3C B#16#3D B#16#3E B#16#3F OB121_FLT_REG	调用 FC 时的 FC 编号错误 调用 FB 时的 FB 编号错误 访问未下载的 DB，DB 编号在允许范围 访问未下载的 FC，FC 编号在允许范围 访问未下载的 SFC，SFC 编号在允许范围 访问未下载的 FB，FB 编号在允许范围 访问未下载的 SFB，SFB 编号在允许范围 非法的编号

有编程错误时，打开 DB121。单击工具栏上的按钮，启动监控功能。图 7-24 是 OB121 的局部变量的前 12B。

地址	名称	类型	初始值	实际值
0.0	ARY[1]	DWORD	DW#16#0	DW#16#25220179
4.0	ARY[2]	DWORD	DW#16#0	DW#16#8C240004
8.0	ARY[3]	DWORD	DW#16#0	DW#16#00010002

图 7-24　OB121 的局部变量

DBB1（OB121_SW_FLT）是错误代码，由 OB121 的在线帮助和表 7-1 可知，16#22 表示读取数据时的区域长度错误。

DBB4（OB121_BLK_TYPE）是错误所在的块的类型，由在线帮助可知 8C 表示 DB。

DBB5（OB121_RESERVED_1）的值为 16#24，由表 7-1 可知，第 4～7 位的 2 表示访问类型为字。第 0～3 位的 4 表示存储器区为共享数据块。

DBW6（错误源，OB121_FLT_REG）的地址为 4，即程序中读操作出错的 DBW4 的地址。

4．编程错误的仿真练习

执行 BCD 码转换为整数的指令 BTI 时，如果 BCD 码的低 3 位中某位为无效数据 16#A～16#F（对应的十进制数为 10～15），转换指令将会出错。

用新建项目向导生成一个项目，可以选任意型号的 CPU。

生成 OB35，编写下面的语句表程序，被转换的常数 W#16#23A 的最低位不是 BCD 码：

```
L      W#16#23A
BTI
T      MW      8
```

打开 PLCSIM，下载用户程序。将仿真 CPU 切换到 RUN-P 模式。由于 OB35 中的编程错误，不能切换到 RUN 模式，CPU 视图对象上的 SF LED 亮。

在 SIMATIC 管理器中执行菜单命令“PLC”→“诊断/设置”→“模块信息”，打开模块信息对话框，用“诊断缓冲区”检查出错的原因。

选中其中的事件“BCD 转换错误”，单击“打开块”按钮，打开出错的块。

打开“堆栈”选项卡，观察 B 堆栈、L 堆栈和 I 堆栈中的内容。

生成 OB121 和 DB121，在 DB121 中生成 20B 的数组 ARY。在 OB121 中将 MW10 加 1，调用 SFC20，用 DB121.ARY 保存 OB121 的局部变量。

下载用户程序后切换到 RUN-P 模式，观察运行的结果。

打开 DB121，单击工具栏上的按钮，启动监控功能。打开 OB121 的在线帮助，分析 DB121 中的 OB121 的局部变量的意义。

7.4 练习题

1．为了防止出现 DP 网络故障时 CPU 进入 STOP 模式，S7-300 和 S7-400 分别需要生成和下载哪些组织块？它们各有什么作用？

2．使用 STEP 7 的在线诊断功能需要什么必要条件？

3．怎样打开快速视图和诊断视图？

4．怎样打开 CPU、DP 从站和从站中的模块的模块信息对话框？

5．怎样获取 OB 的在线帮助信息？

6．根据 OB82 的在线帮助，分析图 7-25 中 OB82 的局部变量的意义。

7．根据 OB86 的在线帮助，分析图 7-26 中 OB86 的局部变量的意义。

地址	名称	类型	初始值	实际值
0.0	ARY[0]	DWORD	DW#16#0	DW#16#39421A52
4.0	ARY[1]	DWORD	DW#16#0	DW#16#C55407FD
8.0	ARY[2]	DWORD	DW#16#0	DW#16#05230000

图 7-25　OB82 的局部变量

地址	名称	类型	初始值	实际值
0.0	ARY[0]	DWORD	DW#16#0	DW#16#39C41A56
4.0	ARY[1]	DWORD	DW#16#0	DW#16#C05407FF
8.0	ARY[2]	DWORD	DW#16#0	DW#16#07FE0103

图 7-26　OB86 的局部变量

第 8 章

人机界面的组态与仿真

8.1 人机界面的硬件与组态软件

8.1.1 人机界面与触摸屏

1．人机界面

人机界面（Human Machine Interface）简称为 HMI。从广义上说，人机界面泛指计算机（包括 PLC）与操作人员交换信息的设备。在控制领域，人机界面一般特指用于操作人员与控制系统之间进行对话和相互作用的专用设备。人机界面可以在恶劣的工业环境中长时间连续运行，是 PLC 的最佳搭档。

人机界面可以用字符、图形和动画动态地显示现场数据和状态，操作人员可以通过人机界面来控制现场的被控对象。此外人机界面还有报警、用户管理、数据记录、趋势图、配方管理、显示和打印报表等功能。

2．触摸屏

触摸屏是人机界面的发展方向，用户可以在触摸屏的屏幕上生成满足自己要求的触摸式按键。触摸屏使用直观方便，易于操作。画面上的按钮和指示灯可以取代相应的硬件元件，减少 PLC 需要的 I/O 点数，降低系统的成本，提高设备的性能和附加价值。

STN 液晶显示器支持的彩色数有限（例如 8 色或 16 色），被称为“伪彩”显示器。STN 显示器的图像质量较差，可视角度较小，但是功耗小、价格低，用于要求较低的场合。

TFT 液晶显示器又称为“真彩”显示器，每一液晶象素点都用集成在其后的薄膜晶体管来驱动，其色彩逼真、亮度高、对比度和层次感强、反应时间短、可视角度大，但是耗电较多，成本较高，用于要求较高的场合。

3．人机界面的工作原理

人机界面最基本的功能是显示现场设备（通常是 PLC）中位变量的状态和寄存器中数字变量的值，用监控画面上的按钮向 PLC 发出各种命令，以及修改 PLC 存储区的参数。

（1）对监控画面组态

首先需要用计算机上运行的组态软件对人机界面组态（见图 8-1）。使用组态软件，可以很容易地生成满足用户要求的人机界面的画面，用文字或图形动态地显示 PLC 中位变量的状态和数字量的数值。用各种输入方式，将操作人员的位变量命令和数字设定值传送到 PLC。画面的生成是可视化的，一般不需要用户编程，组态软件的使用简单方便，很容易掌握。

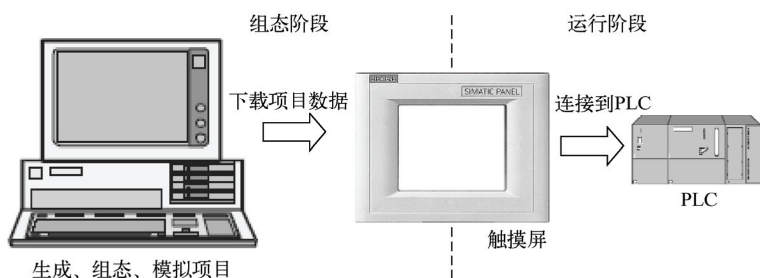


图 8-1 人机界面的工作原理

（2）编译和下载项目文件

编译项目文件是指将建立的画面及设置的信息转换成人机界面可以执行的文件。编译成功后，需要将可执行文件下载到人机界面的存储器。

（3）运行阶段

在控制系统运行时，人机界面和 PLC 之间通过通信来交换信息，从而实现人机界面的各种功能。只需要对通信参数进行简单的组态，就可以实现人机界面与 PLC 的通信。将画面中的图形对象与 PLC 的存储器地址联系起来，就可以实现控制系统运行时 PLC 与人机界面之间的自动数据交换。

8.1.2 SIMATIC 人机界面

西门子的人机界面及其组态软件已升级换代，过去的 170、270、370 系列已被 177、277、377 系列取代，组态软件 Protool 已被 WinCC flexible 取代。

1．SIMATIC 人机界面

SIMATIC HMI 的品种非常丰富，下面是各类 HMI 产品的主要特点：

1）KTP 精简系列面板：具有基本的功能，经济实用，有很高的性能价格比。显示器尺寸有 3.8in、5.7in、10.4in 和 15.1in 这 4 种规格。

2）微型面板：与 S7-200 配合使用，显示器均为单色。有文本显示器 TD 400C，3in 的 OP 73 micro、5.7in 的 TP 177 micro 和 K-TP 178micro。

3）77 系列面板：显示器均为单色，包括 3in 的 OP 73、4.5in 的 OP 77A 和 OP 77B。

4）TP/OP 177/277 系列面板：TP 是触摸面板（触摸屏）的简称，OP 是操作员面板的简称，OP 有多个密封薄膜按键。

TP 177A 采用 5.7in 单色显示器，TP 177B 的显示器有 4.3in 和 5.7in 两种规格，OP 177B 的显示器为 5.7in。TP/OP 277 使用 5.7in 的彩色显示器。

5）MP 177/277/377 系列多功能面板：是功能最强的人机界面，显示器均为 64K 色，MP 177 的显示器为 5.7in。MP 277 有 7.5in、10.4in 显示器的 TP、OP。MP 377 有 10.4in 显示器的 TP、OP，和 15.1in、19in 显示器的 TP。

6）移动面板：移动面板可以在不同的地点灵活应用。Mobile Panel 177 的显示器为 5.7in，Mobile Panel 277 的显示器有 7.5in 和 10.4in 两种规格。

2．WinCC flexible

WinCC flexible 的中文版是免费的，可以组态所有的 SIMATIC 操作面板。用于组态计算机（基于 PC 的工作站）时，需要使用 WinCC flexible 亚洲高级版或英文高级版。

西门子的全集成自动化（TIA）是指控制系统使用统一的通信协议、统一的数据库和统



一的编程组态工具。WinCC flexible 是 TIA 的重要组成部分，它可以与西门子的工业控制软件 STEP 7、iMap 和 Scout 集成在一起。

WinCC flexible 具有开放简易的扩展功能，带有 Visual Basic 脚本功能，集成了 ActiveX 控件，可以将人机界面集成到 TCP/IP 网络。

WinCC flexible 简单、高效，易于上手，功能强大。在创建工程时，通过单击鼠标便可以生成 HMI 项目的基本结构。基于表格的编辑器简化了对象（例如变量、文本和信息）的生成和编辑。通过图形化配置，简化了复杂的配置任务。

WinCC flexible 带有丰富的图库，提供大量的图形对象供用户使用。

WinCC flexible 可以方便地移植原有的 ProTool 项目，支持多语言组态和多语言运行。还可以方便地与 STEP 7 集成，与 PLCSIM 一起实现 PLC 和人机界面的集成仿真。

SIMATIC HMI 的组态和使用方法请参阅参考文献[2]。

8.1.3　安装 WinCC flexible

1．安装 WinCC flexible 的计算机的推荐配置

WinCC flexible 支持所有兼容 IBM/AT 的个人计算机。下面是安装 WinCC flexible 2008 要求的系统配置。

- 操作系统：Windows XP Professional SP2 或 SP3，Windows Vista。
- 图形/分辨率：1024×768 或更高，16 位色。
- 处理器：最低配置为 Pentium Ⅵ或不小于 1.6 GHz 的处理器。
- 主内存（RAM）：最小 1GB（Windows XP）或 1.5GB（Windows Vista），推荐 2 GB。

2．安装 WinCC flexible

双击随书光盘的“WinCC flexible 2008”文件夹中的 Setup.exe，单击各对话框的“下一步”按钮，进入下一对话框。

在“许可证协议”对话框，选中“我接受上述许可证协议……”。

在“要安装的程序”对话框（见图 8-2），确认要安装的软件（用打勾表示），可采用默认的设置。已安装的软件左边的复选框（小方框）为灰色。



图 8-2　要安装的程序

如果要修改安装的路径，选中某个要安装的软件，出现默认的安装文件夹。单击“浏览”按钮，用打开的对话框修改安装的文件夹。建议将该软件安装在 C 盘默认的文件夹。

开始安装软件时出现图 8-3 中的对话框，该对话框不会显示已经安装的软件。

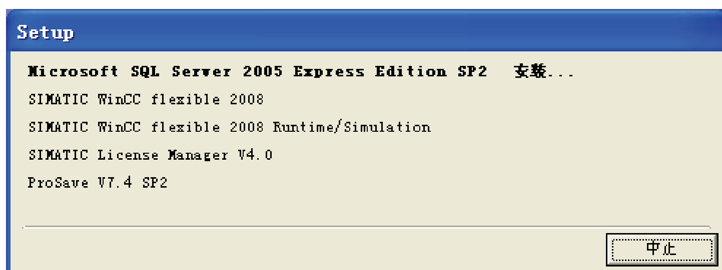


图 8-3　安装过程中显示的对话框

安装过程是自动完成的，不需要用户干预。安装完成后，出现的对话框显示“安装程序已在计算机上成功安装了软件”，单击“完成”按钮，立即重新启动计算机。也可以用单选框选择以后重启计算机。

3．安装软件时遇到的问题的处理

在安装 WinCC flexible 时，可能出现提示“Please restart Windows before installing new programs”（安装新程序之前，请重新启动 Windows）或类似的信息，即使重新启动计算机后再安装软件，还是出现上述信息。可以用 1.2.1 节的安装注意事项中介绍的方法解决这一问题。

8.2　实训四十五　人机界面的画面组态与仿真实验

8.2.1　创建项目和 HMI 站点

1．在 SIMATIC 管理器中创建 HMI 站点

打开 STEP 7，用“新建项目向导”生成一个名为“PLC_HMI”的项目（见随书光盘中的同名例程），CPU 的型号为 CPU 315-2DP。

选中管理器左侧的项目窗口最上面的项目图标，执行菜单命令“插入”→“站点”→“SIMATIC HMI Station”（人机界面站），在出现的对话框中选择 HMI（人机界面）的型号（见图 8-4），单击“确定”按钮，生成的 HMI 站对象出现在 SIMATIC 管理器的项目窗口中（见图 8-5）。

创建 HMI 站点实际上就是创建集成在 STEP 7 中的 WinCC flexible 项目，后者保存在 STEP 7 项目的“HmiEs”文件夹中。

在创建 WinCC flexible 的新项目时，如果出现错误提示“Could not find file "C:\Documents and Settings\All Users\Application Data\Siemens AG\SIMATIC WinCC flexible 2008\Caches \1.3.0.0_ 183.1\Read\Template_zh-CN.tmp"”，可能是用优化大师或 360 卫士之类的系统工具清除垃圾文件时，自动删除了临时文件*.tmp 引起的。将文件夹“C:\Documents and Settings \All Users \Application Data \Siemens AG \SIMATIC WinCC flexible 2008”删除，然后重新创建 Wincc flexible 的项目。上述文件夹会在 flexible 软件再次启动时重新创建，软件就能正常



使用了。

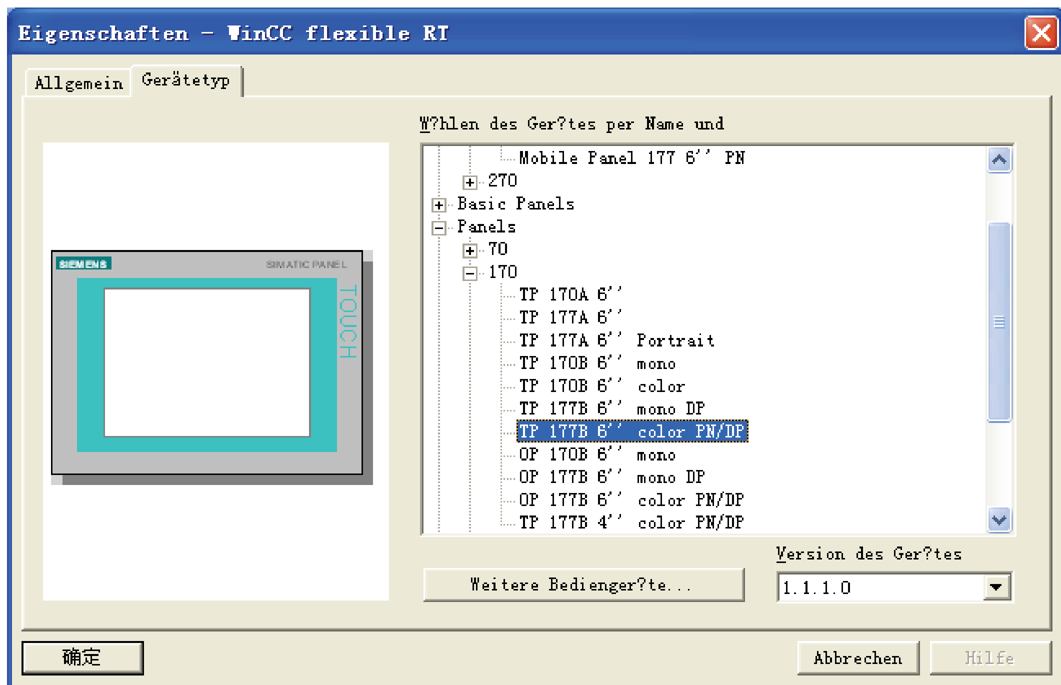


图 8-4　设置 HMI 的型号

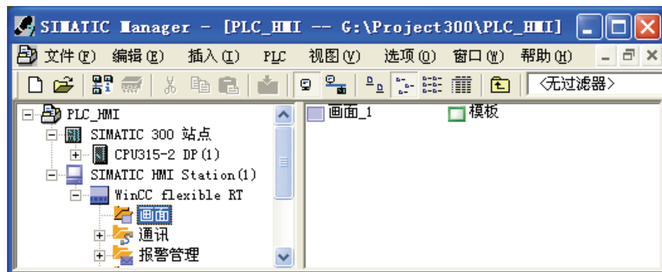


图 8-5　集成了 HMI 站的 STEP 7 项目

2．建立两个站点的连接

为了实现 PLC 和 HMI 设备之间的自动数据交换，需要建立它们之间的连接。单击工具栏上的按钮，打开网络组态工具 NetPro（见图 8-6 上面的图），显示出 STEP 7 项目中尚未与 MPI 网络连接的 S7-300 站点和 HMI 站点。它们的 MPI 接口用红色的小方块表示，MPI 的站地址分别为 2 和 1。

双击 MPI 接口对应的小红方块，打开 MPI 接口属性对话框（见图 8-6 下面的图），选中“子网”列表中的 MPI 网络，单击“确定”按钮，站点就被连接到 MPI 网络上了。可以用该对话框的“地址”选择框修改站点的 MPI 地址。

也可以将 MPI 接口对应的小红方块直接拖放到红色的 MPI 网络上。

图 8-6 是已经连接好的网络图。单击工具栏上的按钮，保存和编译网络组态信息。

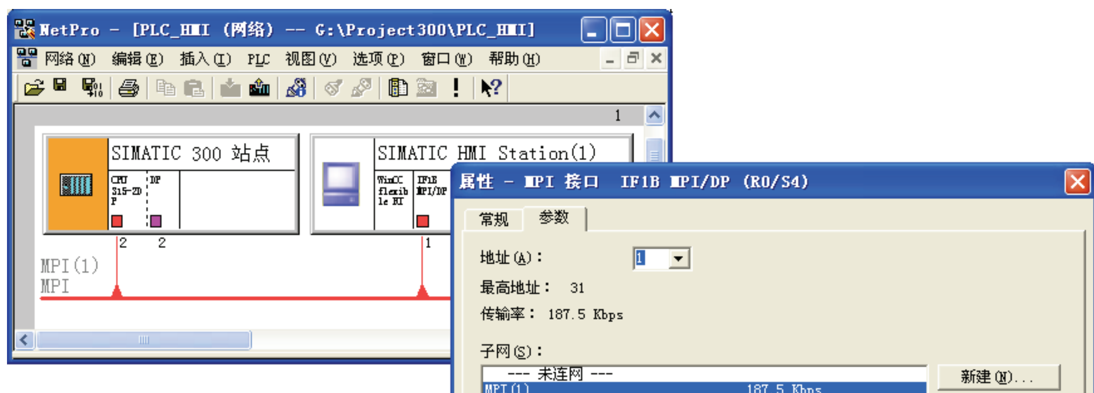


图 8-6　用 NetPro 组态 MPI 网络连接

3．定义符号和编写程序

在 PLC 的符号表中生成 5 个符号地址（见图 8-7），M0.0 和 M0.1 是面板画面上的起动按钮和停车按钮提供给 PLC 的控制信号。图 8-8 是 OB1 中的梯形图程序，其中的 MOVE 指令只是用来检验 HMI 设备与仿真 PLC 之间的数据交换，没有什么实际的意义。

符号编辑器 - [S7 程序(1) (符号) -...]

	状态	符号	地址		数据类型	注释
1		起动按钮	M	0.0	BOOL	
2		停止按钮	M	0.1	BOOL	
3		液位设定值	MW	2	INT	
4		液位测量值	MW	4	INT	
5		水泵	Q	0.0	BOOL	

图 8-7　STEP 7 中的符号表

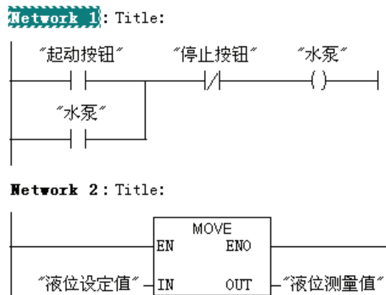


图 8-8　OB1 中的梯形图

4．打开 WinCC flexible 的项目

返回 SIMATIC 管理器，选中 WinCC flexible 站点中的“画面”（见图 8-5），双击管理器右侧的工作区中的“画面_1”图标，自动打开 WinCC flexible 的“画面”编辑器（见图 8-9）。

打开画面后，可以使用工具栏上的放大按钮和缩小按钮，来放大或缩小画面。

在画面编辑器下面的属性对话框中，可以设置画面的名称和编号。单击“背景色”选择框的键，在出现的颜色列表中设置画面的背景色为白色。

5．工作区的操作

用户在工作区编辑项目对象，除了工作区，可以对其他窗口（例如项目视图和工具箱等）进行移动、改变大小和隐藏等操作。同时打开多个编辑器时，单击图 8-9 中编辑器工作区上面的标签，可以显示对应的编辑器。如果不能全部显示被同时打开的编辑器的标签，可以用和按钮来左右移动编辑器的标签。单击工作区右上角的按钮，将会关闭当前显示的编辑器。

打开对象视图，用鼠标调整它的大小和位置。选中项目视图中的“变量”，对象视图出



现变量表中的变量。

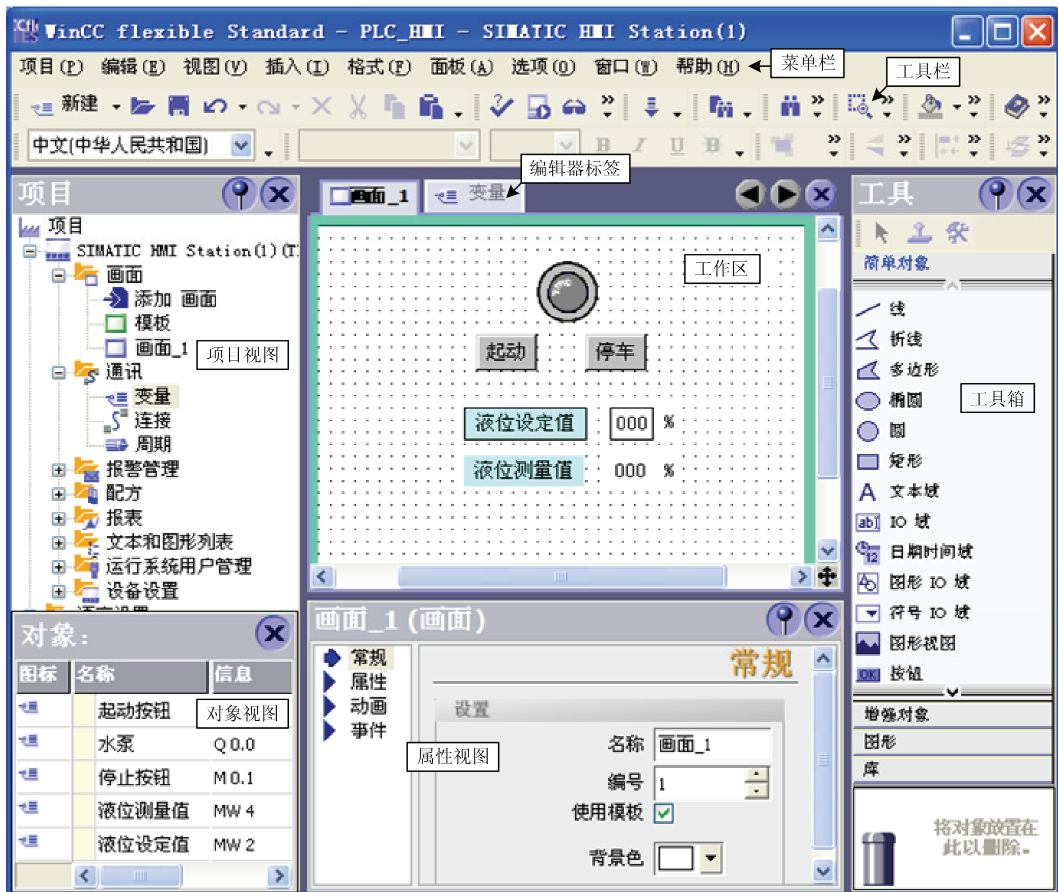


图 8-9　WinCC flexible 的用户界面

6．打开连接

双击右侧项目视图中的“连接”，打开连接编辑器（见图 8-10）。连接表内自动生成了在 NetPro 中组态的连接，其默认的名称为“连接_1”，通信对象为 S7-300/400，在“激活的”列选中“开”。连接表下面的属性视图给出了通信连接的参数。



图 8-10　连接编辑器

8.2.2　组态指示灯

1．打开库文件

工具箱中没有用于显示位变量 ON/OFF 状态的指示灯，下面介绍使用对象库中的指示灯

的方法。

选中工具箱中的“库”文件夹（见图 8-11），用右键单击下面的空白区，在弹出的快捷菜单中执行命令“库…”→“打开”。在出现的对话框中，打开文件夹“C:\Program Files\Siemens\SIMATIC WinCC flexible\WinCC flexible Support\Libraries\System-Libraries”，双击打开按钮与开关库文件“Button_and_switches.wlf”。

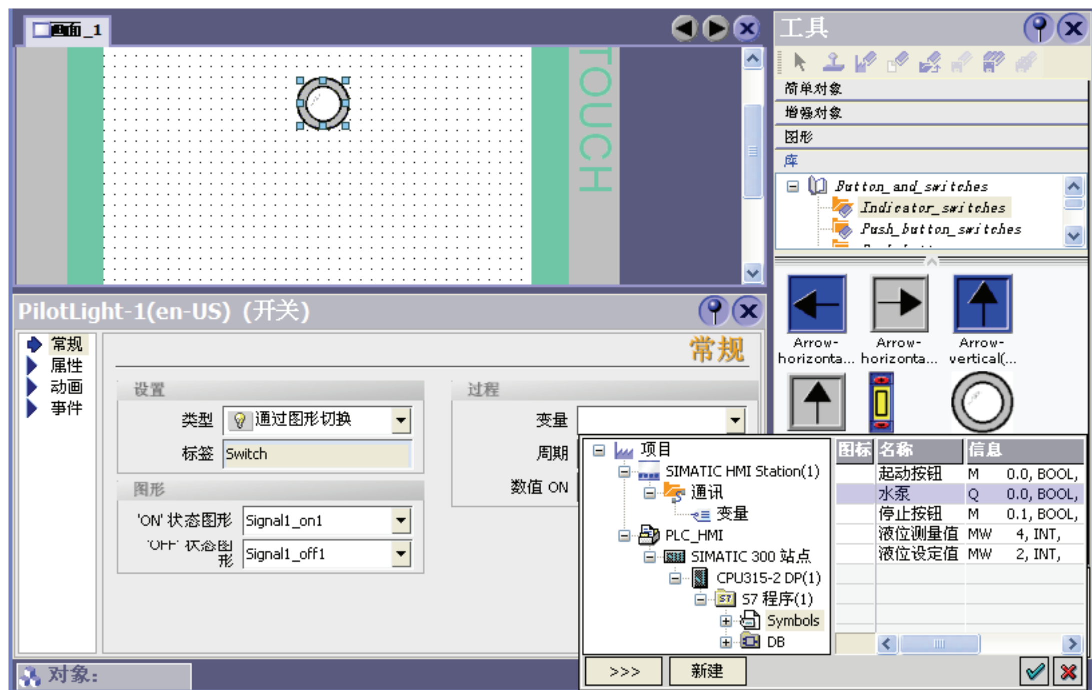


图 8-11　组态指示灯连接的变量

2. 生成指示灯

打开刚刚装入的“Button_and_switches”库（见图 8-11），选中该库中的 Indicator_switches（指示灯/开关）。

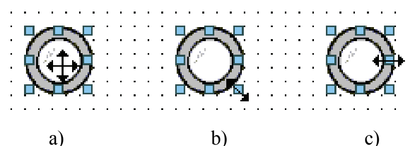
用鼠标左键按住其中的指示灯不放，同时移动鼠标，未移到画面上时鼠标的光标为（禁止放置），移动到画面上时，鼠标的光标变为（可以放置）。

在画面上的适当位置放开鼠标左键，指示灯被放置到画面上当时所在的位置。此时指示灯的四周有 8 个小矩形，表示处于被选中的状态。

3. 用鼠标改变对象的位置和大小

用鼠标左键单击图 8-12a 的指示灯，它的四周出现 8 个小正方形。将鼠标的光标放到指示灯上，光标变为图中的十字箭头图形。按住鼠标左键并移动鼠标，将选中的对象拖到希望的位置。松开左键，对象被放在该位置。

用鼠标左键选中某个角的小正方形，鼠标的光标变为 45°的双向箭头（见图 8-12b），按住左键并移动鼠标，可以同时改变对象的长度和宽度。



a)　b)　c)

图 8-12　对象的移动与缩放



用鼠标左键选中 4 条边中点的某个小正方形，鼠标的光标变为水平或垂直的双向箭头（见图 8-12c），按住左键并移动鼠标，可将选中的对象沿水平方向或垂直方向放大或缩小。

4. 组态指示灯连接的变量

选中画面上的指示灯，画面下面是指示灯的属性视图（见图 8-11）。属性视图左侧有一个树形结构，可以用它来选择各种属性类别。双击画面编辑器中的对象，可以打开或关闭它的属性视图。

用属性视图左侧窗口向右的箭头表示被选中的属性组，例如图 8-11 中的“常规”属性组。属性视图的右侧区域用于对当前所选属性组进行组态。其中的“常规”组用来设置最重要的属性。

将 WinCC flexible 的项目集成到 STEP 7 的项目中后，对画面对象组态时，可以直接使用 STEP 7 的符号表或数据块中的符号地址。

选中属性视图中的“常规”组，单击右边的“变量”选择框右侧的▾按钮（见图 8-11），在出现的变量列表中，打开 STEP 7 中 S7-300 站的符号表（Symbols），单击其中的变量“水泵”，该变量出现在显示框，就建立起了该变量与指示灯的连接关系，即用指示灯显示变量“水泵”的状态。

双击图 8-9 的项目视图中的“变量”，打开变量编辑器，可以看到自动生成的组态时使用的 STEP 7 中的变量。

5. 指示灯图形的组态

指示灯分别用图形 Signal1_on1 和 Signal1_off1 来表示指示灯的点亮（对应的变量为 1 状态）和熄灭（对应的变量为 0 状态）状态（见图 8-11）。

图形 Signal1_on1 的中间部分为深色，图形 Signal1_off1 的中间部分为浅色（见图 8-13）。一般用浅色表示指示灯点亮，所以需要用下面的操作来交换属性视图中两个状态的图形。

单击图 8-11 的属性视图中“‘ON’状态图形”选择框右侧的▾按钮，选中出现的图形列表（见图 8-13）中的“Signal1_off1”，窗口的右侧出现选中的指示灯图形。单击“设置”按钮，关闭图形列表。这样“ON”状态（变量为 1 状态）的指示灯图形的中间部分变为浅色。用同样的方法，设置“OFF”状态（0 状态）指示灯的图形为 Signal1_on1，中间部分为深色。



图 8-13 图形列表

8.2.3 组态按钮

1. 按钮的生成

画面上的按钮与接在 PLC 输入端的物理按钮的功能相同，用来将操作命令发送给 PLC，通过 PLC 的用户程序来控制生产过程。

单击工具箱中的“简单对象”组，将其中的按钮图标拖放到画面上，放开鼠标左键，按钮被放置在画面上。可以用前面介绍的鼠标的使用方法来调整按钮的位置和大小。

2. 设置按钮的属性

选中生成的按钮，在属性视图的“常规”对话框中（见图 8-14），用单选框设置“按钮模式”和“文本”均为“文本”。

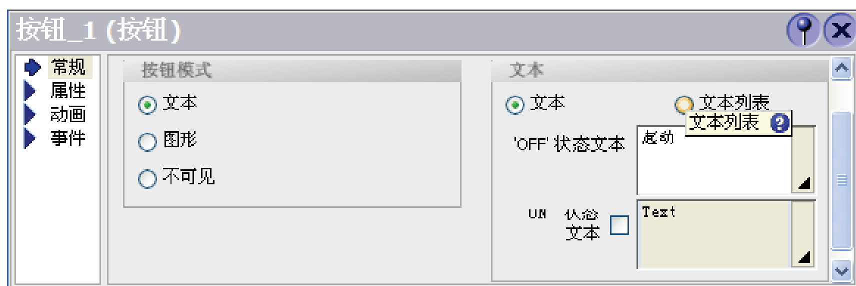


图 8-14　组态按钮的常规属性

如果选中复选框“ON 状态文本”，可以分别设置按下和释放按钮时，按钮上面的文本。未选中该复选框时，按钮按下和释放时显示的文本相同。

选中图 8-15 左边窗口的“外观”组，可以在右边窗口修改它的背景色和文本的颜色。还可以用复选框设置按钮是否有三维效果。



图 8-15　组态按钮的外观

在运行时如果单击了该按钮，它称为焦点。按钮上的字符“起动”周围出现一个虚线的方框（见图 8-22）。图 8-15 中的焦点颜色用来设置表示焦点的虚线框的颜色，一般采用默认的黑色。

在属性视图的“布局”对话框（见图 8-16），可以设置对象的位置和大小。一般在画面上直接用鼠标设置画面元件的位置和大小，这样比在“布局”对话框中修改参数更为直观。



如果选中复选框“自动调整大小”，将根据按钮上的文本字数和字体的大小自动调整按钮的大小。可以用“布局”对话框微调按钮的位置和大小。

选中左边窗口的“文本”组（见图 8-17），可以定义按钮上文本的字体、大小和对齐方式。

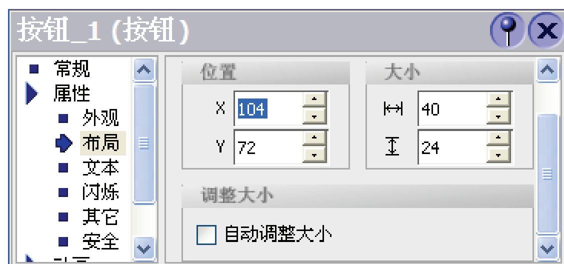


图 8-16　组态按钮的布局



图 8-17　组态按钮的文本格式

3. 按钮功能的设置

在属性视图的“事件”类的“按下”对话框（见图 8-18），单击视图右侧最上面一行，再单击它的右侧出现的▼键（在单击之前它是隐藏的），单击出现的系统函数列表的“编辑位”文件夹中的函数“SetBit”（置位）。

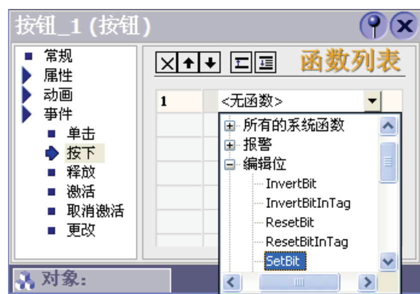


图 8-18　组态按钮按下时执行的函数

直接单击表中第 2 行右侧隐藏的▼按钮，打开出现的对话框中 PLC 的符号表，单击其中的变量“起动按钮”（M0.0，见图 8-19）。在运行时按下该按钮，将变量“起动按钮”置位为 1 状态。

用同样的方法，设置在释放按钮时调用系统函数“ResetBit”，将变量“起动按钮”复位为 0 状态。该按钮具有点动按钮的功能，按下按钮时变量“起动按钮”被置位，放开按钮时它被复位。

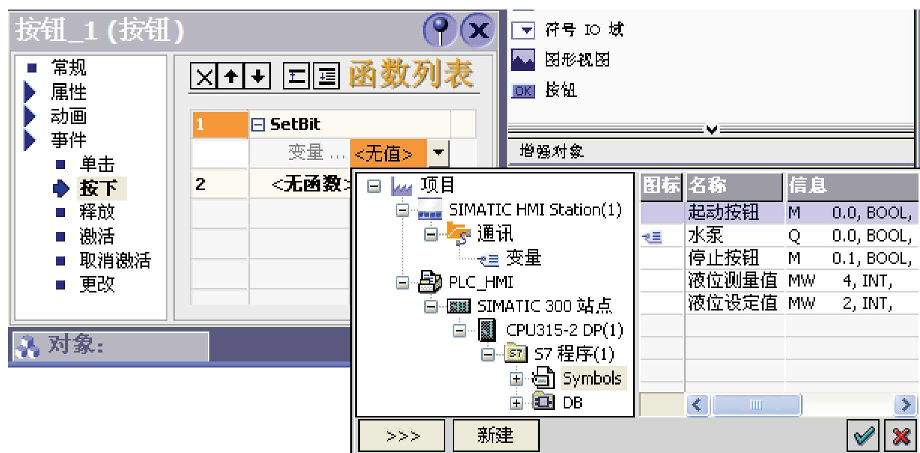


图 8-19　组态按钮按下时操作的变量

单击画面上组态好的起动按钮，先后执行“编辑”菜单中的“复制”和“粘贴”命令，生成一个相同的按钮。用鼠标调节它的位置，选中属性视图的“常规”组，将按钮上的文本修改为“停车”。选中“事件”组，组态在按下和释放按钮时分别将变量“停止按钮”置位和复位。

8.2.4 组态文本域与 IO 域

1. 生成与组态文本域

将工具箱中标有“A”的文本域图标（见图 8-9）拖放到画面上，默认的文本为“Text”。选中生成的文本域，打开属性视图中的“常规”对话框，在右边的文本框中输入文本“液位设定值”。

选中属性视图左边窗口的“外观”组，可以在右边窗口（见图 8-20）修改它的文本的颜色、背景色和填充样式。“边框”域中的“样式”选择框可以选择“无”（没有边框）（见图 8-9）和“实心”（有边框），还可以设置边框的宽度和颜色，用复选框设置是否有三维效果。



图 8-20 组态文本域的外观

“布局”的设置与按钮的基本上相同（见图 8-16），还可以设置四周的边距。“文本”的设置与按钮的基本上相同（见图 8-17），还可以用“方向”选择框设置文本域是水平放置或垂直放置。

选中画面上生成的文本域，执行复制和粘贴操作，将生成的文本域的文字改为“液位测量值”，没有边框（见图 8-9）。

2. 生成与组态 IO 域

IO 域有 3 种模式：

1）输出域：用于显示变量的数值。

2）输入域：用于操作员输入要传送到 PLC 的数字、字母或符号，将输入的数值保存到指定的变量中。

3）输入/输出域：同时具有输入和输出功能，操作员可以用它来修改 PLC 中变量的数值，并将修改后 PLC 中的数值显示出来。

将工具箱中的 IO 域图标 ab| （见图 8-9）拖放到画面上，选中生成的 IO 域后，选中属性视图左边窗口的“常规”组（见图 8-21），用“模式”选择框设置 IO 域为输入域，连接的变量为“液位设定值”。

在“格式”域，显示类型为默认的“十进制”，“格式样式”被自动设置为 999（3 位整数）。





图 8-21　组态 IO 域

IO 域属性视图中的“外观”、“布局”和“文本”的参数设置与文本域的基本上相同。

选中画面上生成的 IO 域，执行复制和粘贴操作。将新生成的 IO 域放置好后选中它，在属性视图的“常规”对话框，设置该 IO 域连接的变量为“液位实际值”，模式改为输出，没有边框，其余的参数不变。

在两个 IO 域的右边，生成文本域“%”。

8.2.5　PLC 与人机界面的集成仿真实验

WinCC flexible 提供了一个模拟器软件，在没有 HMI 设备的情况下，可以用 WinCC flexible 的运行系统来模拟 HMI 设备，用它来测试项目，调试已组态的 HMI 设备的功能。模拟调试也是学习 HMI 设备的组态方法和提高动手能力的重要途径。有下列 3 种模拟调试的方法：

1．不带控制器连接的模拟（离线模拟）

如果手中既没有 HMI 设备，也没有 PLC，可以用离线模拟功能来检查人机界面的部分功能。可以用模拟表来设置和显示 PLC 中变量的数值。因为没有运行 PLC 的用户程序，离线模拟只能模拟人机界面的部分功能，例如画面的切换和数据的输入过程。

2．带控制器连接的模拟（在线模拟）

设计好 HMI 设备的画面后，如果没有 HMI 设备，但是有 PLC，可以用通信适配器或通信处理器连接计算机和 S7-300/400 的通信接口，进行在线模拟，用计算机模拟 HMI 设备的功能。因为不用将组态信息下载到 HMI 设备，节约了调试时间。在线模拟的效果与实际系统基本上相同。

3．集成模式下的模拟（集成模拟）

将 WinCC flexible 的项目集成在 STEP 7 的项目中，用 WinCC flexible 的运行系统来模拟 HMI 设备，用 S7-300/400 的仿真软件 S7-PLCSIM 来模拟 PLC。这种模拟不需要 HMI 设备和 PLC 的硬件，模拟的效果与实际系统的运行情况基本上相同。

下面是对项目“PLC_HMI”作集成模拟的步骤：

1）在 SIMATIC 管理器中，打开 PLCSIM。将用户程序和组态信息下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。

2）单击 WinCC Flexible 工具栏上的按钮，起动运行系统，开始模拟 HMI。

3）单击画面上的“起动”按钮（见图 8-22），PLCSIM 中的 MB0 视图对象的第 0 位出现“√”，表示 M0.0 被置为 1，由于图 8-8 中的梯形图程序的作用，Q0.0（水泵）变为 1 状态，画面上的指示灯亮。放开起动按钮，M0.0 变为 0 状态，视图对象中的“√”消失。

单击画面上的停车按钮，M0.1 变为 1 状态，指示灯熄灭。

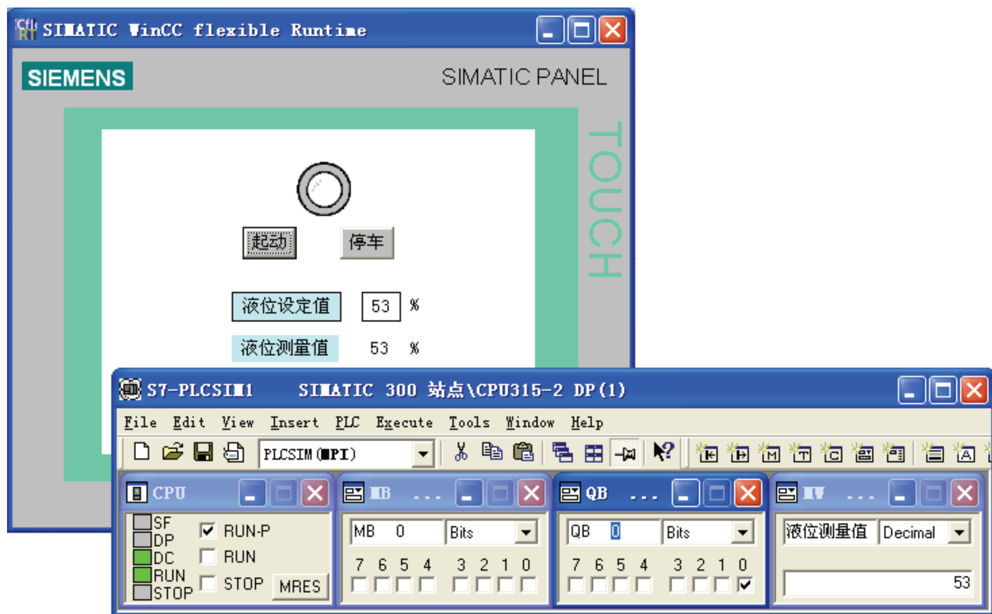


图 8-22　PLC 与 HMI 的集成仿真

4）单击画面上的文本域“液位设定值”右侧的输入域，画面上出现一个数字键盘（见图 8-23）。其中的“ESC”是取消键，单击它后数字键盘消失，退出输入过程，输入的数字无效。“BSP”是退格键，与计算机键盘上的〈Backspace〉键的功能相同，单击该键，将删除光标左侧的数字。“+/-”键用于改变输入的数字的符号。←和→分别是光标左移键和光标右移键，↵是确认（回车）键，单击它使输入的数字有效（被确认），同时关闭键盘。输入的数字将在输入域中显示。

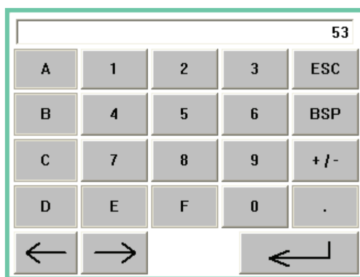


图 8-23　数字键盘

用弹出的小键盘输入数据 53，按回车键确认，由于图 8-8 梯形图程序中 MOVE 指令的作用，变量“液位测量值”（MW4）的值也变为 53。可以看到 PLCSIM 的视图对象和画面上的输出域显示出相同的数值（见图 8-22）。

单击模拟面板右上角的☒按钮，关闭模拟面板。

4．人机界面的集成仿真练习

打开随书光盘中的项目“电机控制”，生成一个 HMI 站点，HMI 的型号为 TP 177B 6” color PN/DP。将两个站点连接到 MPI 网络上。打开 WinCC flexible，在“连接”编辑器打开 PLC 和 HMI 的连接。在画面上生成两个指示灯，用来显示电机正转和反转接触器的状态。在指示灯下面生成文本域，显示指示灯的功能。生成 3 个按钮，分别控制电机的正转、反转和停车。

打开 PLCSIM，将用户程序和组态信息下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模



式。单击 WinCC Flexible 工具栏上的按钮，起动运行系统，用本实训介绍的方法进行集成仿真。

8.3　练习题

1．触摸屏有什么优点？
2．STN 和 TFT 液晶显示器各有什么特点？
3．简述人机界面的工作原理。
4．怎样在 STEP 7 的项目中生成 HMI 站对象？
5．怎样建立 WinCC flexible 与 STEP 7 之间的连接？
6．怎样打开在 STEP 7 中集成的 WinCC flexible 项目？
7．怎样组态具有点动功能的按钮？
8．项目文件有哪几种模拟调试方法？各有什么特点？

第 9 章

PID 闭环控制

9.1 PID 闭环控制系统

9.1.1 模拟量闭环控制系统的组成

典型的 PLC 模拟量闭环控制系统如图 9-1 所示, 虚线中的部分是用 PLC 实现的。

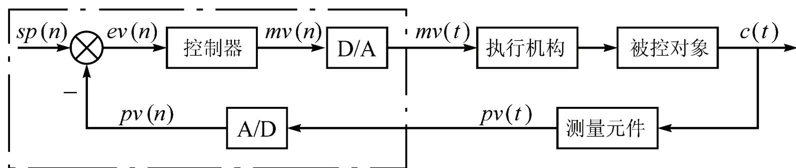


图 9-1 PLC 模拟量闭环控制系统方框图

在模拟量闭环控制系统中，被控量 $c(t)$（例如压力、温度、流量、转速等）是连续变化的模拟量，大多数执行机构(例如电动调节阀和变频器等)要求 PLC 输出模拟信号 $mv(t)$，而 PLC 的 CPU 只能处理数字量。$c(t)$首先被测量元件（传感器）和变送器转换为标准量程的直流电流信号或直流电压信号 $pv(t)$，例如 4～20mA 和 0～10V 的信号，PLC 用模拟量输入模块中的 A/D 转换器将它们转换为数字量 $pv(n)$。

模拟量与数字量之间的相互转换和 PID 程序的执行都是周期性的操作，其间隔时间称为采样周期 T_S 。各数字量括号中的 n 表示该变量是第 n 次采样计算时的数字量。

图 9-1 中的 $sp(n)$是给定值，$pv(n)$为 A/D 转换后的反馈量，误差 $ev(n) = sp(n) - pv(n)$。

D/A 转换器将 PID 控制器输出的数字量 $mv(n)$转换为模拟量（直流电压或直流电流）$mv(t)$，再去控制执行机构。

例如在加热炉温度闭环控制系统中，用热电偶检测炉温，温度变送器将热电偶输出的微弱的电压信号转换为标准量程的电流或电压，然后送给模拟量输入模块，经 A/D 转换后得到与温度成比例的数字量，CPU 将它与温度设定值比较，并按某种控制规律（例如 PID 控制算法）对误差值进行运算，将运算结果 (数字量) 送给模拟量输出模块，经 D/A 转换后变为电流信号或电压信号，用来控制电动调节阀的开度，通过它控制加热用的天然气的流量，实现对温度的闭环控制。

闭环负反馈控制可以使控制系统的反馈量 $pv(n)$等于或跟随给定值 $sp(n)$。以炉温控制系统为例，假设输出的温度值 $c(t)$低于给定的温度值，反馈量 $pv(n)$小于给定值 $sp(n)$，误差 $ev(n)$



为正，控制器的输出量 $mv(t)$将增大，使执行机构（电动调节阀）的开度增大，进入加热炉的天然气流量增加，加热炉的温度升高，最终使实际温度接近或等于给定值。

天然气压力的波动、常温的工件进入加热炉，这些因素称为扰动量，它们会破坏炉温的稳定。闭环控制可以有效地抑制闭环中各种扰动的影响，使被控量趋近于给定值。

闭环控制系统的结构简单，容易实现自动控制，因此在各个领域得到了广泛的应用。

9.1.2　PID 控制器

1．怎样实现 PID 控制

PID 是比例、微分、积分的缩写，PID 控制器是应用最广的闭环控制器。

S7-300/400 的闭环控制模块为 FM335 和 FM335-2，它们集成了闭环控制需要的 I/O 点和软件，可选用 PID 控制器或自整定模糊控制器。

一般使用普通的信号模块和专用的功能块（FB）来实现 PID 控制。所有型号的 CPU 可以使用 FB41～FB43 和用于温度控制的 FB58 和 FB59。CPU 31xC 还可以使用集成在 CPU 中的 SFB41～SFB43，本章介绍应用最广的 FB41 的使用方法和仿真实验。

FB41“CONT_C”（连续控制器）的输出为连续的模拟量。可以用 FB41 作为单独的 PID 控制器，也可以用脉冲发生器 FB43 进行扩展，产生脉冲宽度调制的开关量输出信号，来控制比例执行机构。

用于 PID 控制的功能块的参数很多，建议结合功能块的方框图来学习和理解这些参数。

2．PID 控制器输出的表达式

模拟量 PID 控制器的输出表达式为

$$mv(t) = K_P[ev(t) + \frac{1}{T_I}\int ev(t)dt + T_D \frac{dev(t)}{dt}] + M \qquad (9-1)$$

式中控制器的输入量（误差信号）为

$$ev(t) = sp(t) - pv(t)$$

$sp(t)$为设定值，$pv(t)$为过程变量（反馈值）；$mv(t)$是控制器的输出信号，K_P 为比例系数（或称为增益），T_I 和 T_D 分别是积分时间和微分时间，M 是积分部分的初始值。

式（9-1）中等号右边的前 3 项分别是比例、积分、微分部分，它们分别与误差 $ev(t)$、误差的积分和误差的微分成正比。如果取其中的一项或两项，可以组成 P、PI 或 PD 调节器。需要较好的动态品质和较高的稳态精度时，可以选用 PI 控制方式；控制对象的惯性滞后较大时，应选择 PID 控制方式。

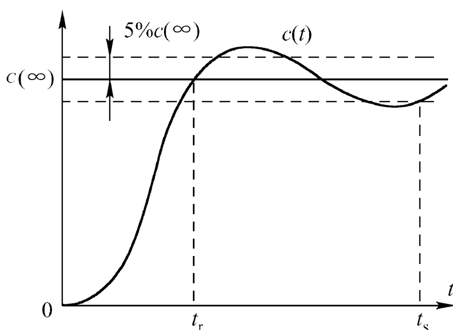


图 9-2　被控对象的阶跃响应曲线

3．闭环控制的主要性能指标

由于给定输入信号或扰动输入信号的变化，系统的输出量达到稳态值之前的过程称为过渡过程或动态过程。系统的动态性能常用阶跃响应曲线（见图 9-2）的参数来描述。阶跃输入信号在 $t = 0$ 之前为 0，$t > 0$ 时为某一恒定值。

输出量第一次达到稳态值的时间 t_r 称为上升时间，上升时间反映了系统在响应初期的快速性。

系统进入并停留在稳态值 $c(\infty)$ 上下±5%（或 2%）的误差带内的时间 t_s 称为调节时间，到达调节时间表示过渡过程已基本结束。

设动态过程中输出量的最大值为 $c_{max}(t)$，如果它大于输出量的稳态值 $c(\infty)$，定义超调量为

$$\sigma\% = \frac{c_{max}(t) - c(\infty)}{c(\infty)} \times 100\%$$

超调量反映了系统的相对稳定性，它越小动态稳定性越好，一般希望超调量小于 10%。

系统的稳态误差是进入稳态后的期望值与实际值之差，它反映了系统的稳态精度。

9.1.3 过程变量的处理

下面介绍 FB41 的内部结构和输入、输出变量的意义。

输入变量 SP_INT（内部设定值）是以百分数为单位的浮点数设定值（见图 9-3）。

1．过程变量的输入

可以用两种方式输入过程变量（即反馈值）：

1）用 PV_IN（过程输入变量）输入浮点数格式的过程变量，此时 Bool 输入变量 PVPER_ON（外部设备过程变量 ON）应为 0 状态。

2）用 PV_PER（外部设备过程变量）输入以百分数为单位的浮点数格式的过程变量，即用模拟量输入模块输出的数字值作为 PID 控制的过程变量，此时 PVPER_ON 应为 1 状态。

2．外部设备过程变量转换为浮点数

在 FB41 内部，PID 控制器的给定值 SP_INT、过程变量 PV_IN 和输出值 LMN 都是用浮点数格式的百分数来表示的。

图 9-3 中的 CRP_IN 方框将外部设备输入值 PV_PER（0～27648 或±27648，对应于模拟量输入的满量程）转换为 0～100%或±100％的浮点数格式的数值，CPR_IN 的输出（以％为单位）用下式计算：

PV_R = PV_PER ×100 /27648（％）

3．外部设备过程变量的格式化

PV_NORM 方框用下面的公式将 CRP_IN 的输出 PV_R 格式化：

PV_NORM 的输出 = PV_R × PV_FAC + PV_OFF

上式中 PV_FAC 为过程变量的系数，默认值为 1.0；PV_OFF 为过程变量的偏移量，默认值为 0.0。PV_FAC 和 PV_OFF 用来调节过程输入的范围。

图 9-3 中的 PV（过程变量）为 FB 输出的中间变量。

9.1.4 PID 控制器的参数

1．误差的计算与死区

用浮点数格式的设定值 SP_INT 减去转换为浮点数格式的过程变量 PV（即反馈值），便得到负反馈的误差。



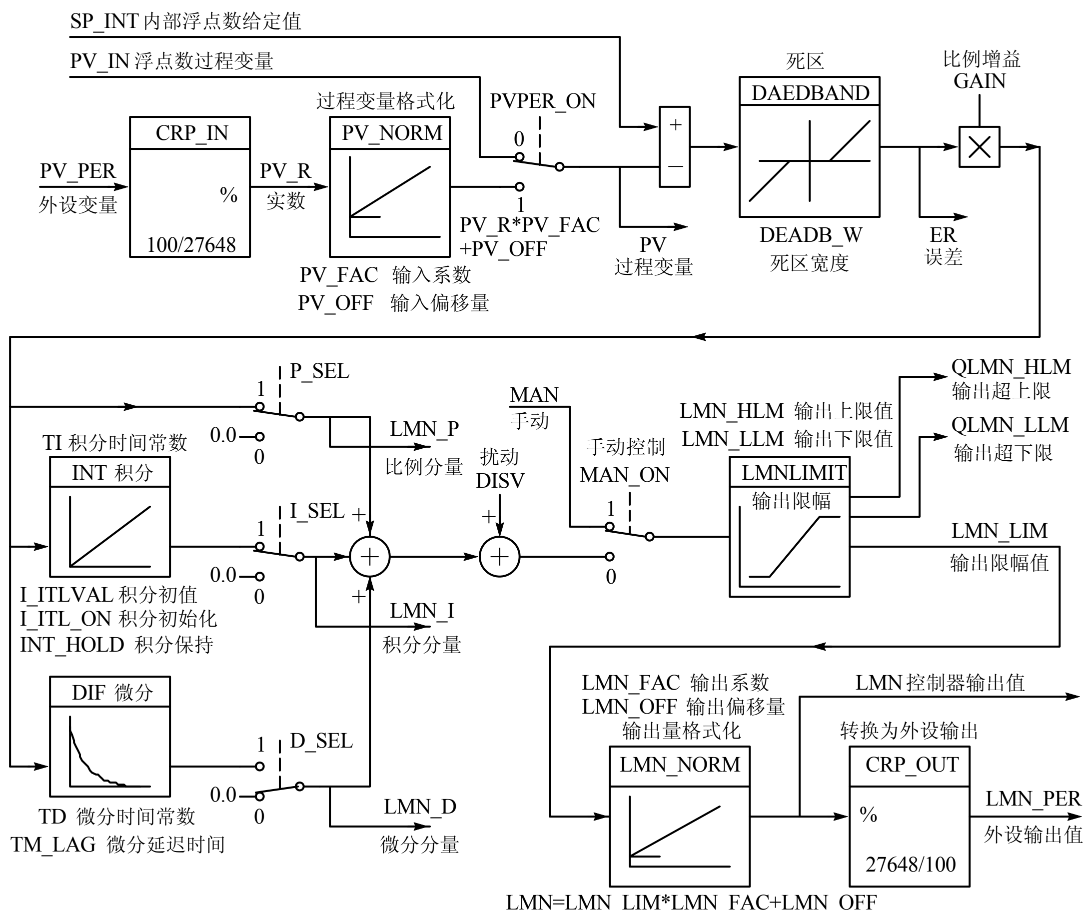


图 9-3　FB41 CONT_C 的框图

在控制系统中，某些执行机构如果频繁动作，将会导致小幅振荡，造成严重的机械磨损。从控制要求来说，很多系统又允许被控量在一定范围内有少量的误差。带死区的 PID 控制器（见图 9-3）能防止执行机构的频繁动作。当死区非线性环节的输入量（即误差）的绝对值小于参数 DEADB_W（死区宽度）时，死区非线性的输出量（即 PID 控制器的输入量）为 0，这时 PID 控制器的输出分量中，比例部分和微分部分为 0，积分部分保持不变，因此 PID 控制器的输出保持不变，控制器不起调节作用，系统处于开环状态。当误差的绝对值超过 DEADB_W 时，开始正常的 PID 控制。如果令 DEADB_W 为 0，死区被关闭。

2．设置控制器的结构

FB41 采用位置式 PID 算法，比例运算、积分运算和微分运算 3 部分并联，可以单独启用或取消它们，因此可以将控制器组态为 P、PI、PD 和 PID 控制器。

P_SEL、I_SEL 和 D_SEL 为 1 时分别启用比例、积分和微分作用，反之则禁止对应的控制作用。默认的控制方式为 PI 控制。

LNM_P、LNM_I 和 LNM_D 分别是 PID 控制器输出量中的比例分量、积分分量和微分分量，它们供调试时使用。

图 9-3 中的 GAIN 为比例部分的增益或比例系数，对应于式（9-1）中的 K_P。TI 和 TD

分别为积分时间和微分时间，对应于式（9-1）中的 T_I 和 T_D。

微分的引入可以改善系统的动态性能，但是也容易引入高频干扰，为此在微分部分增加一阶惯性滤波环节，以平缓输出值的剧烈变化。滤波时间对应于微分延迟时间 TM_LAG，手册建议 TM_LAG = TD/5。

引入扰动量 DISV（Disturbance）可以实现前馈控制，一般设置 DISV 为 0.0。

3．积分器的初始值

FB41 有一个初始化程序，在输入参数 COM_RST（完全重新起动）设置为 1 时该程序被执行。在初始化过程中，如果数字量输入 I_ITL_ON（积分作用初始化）为 1 状态，将输入变量 I_ITLVAL 作为积分器的初始值。

INT_HOLD 为 1 时积分操作保持，积分输出被冻结，一般不冻结积分输出。

9.1.5 控制器输出值的处理

1．手动模式

位变量 MAN_ON 为 1 时为手动模式，为 0 时为自动模式。在手动模式，控制器的输出值被手动输入值 MAN 代替。

2．输出限幅

LMNLIMIT（输出量限幅）方框用于将控制器输出值限幅。

LMNLIMIT 的输入量超出控制器输出的上限值 LMN_HLM 时，输出位 QLMN_HLM（输出超出上限）变为 1 状态；小于下限值 LMN_LLM 时，输出位 QLMN_LLM（输出超出下限）变为 1 状态。LMN_HLM 和 LMN_LLM 的默认值分别为 100.0%和 0.0%。

3．输出量的格式化处理

LMN_NORM（输出量格式化）方框用下述公式来将功能 LMNLIMIT 的输出量 LMN_LIM 格式化：

$$LMN = LMN_LIM \times LMN_FAC + LMN_OFF$$

上式中 LMN 是格式化后浮点数格式的控制器输出值；LMN_FAC 为输出量的系数，默认值为 1.0；LMN_OFF 为输出量的偏移量，默认值为 0.0；LMN_FAC 和 LMN_OFF 用来调节控制器输出量的范围。

4．输出量转换为外部设备（I/O）格式

控制器输出值如果要送给模拟量输出模块中的 D/A 转换器，应使用“CPR_OUT”方框输出的外部设备（I/O）格式的变量 LMN_PER。转换公式为

$$LMN_PER = LMN \times 27648/100$$

9.2 实训四十六 PID 控制的编程与仿真实验

9.2.1 生成项目

PID 控制的难点不是编写或阅读程序，而是整定控制器的参数，需要调试的参数主要有比例系数 GAIN、积分时间 TI、微分时间 TD 和采样周期 CYCLE。如果使用 PI 控制器，也有 3 个主要的参数需要整定。如果参数整定得不好，即使程序设计没有问题，系统的动静态



性能也达不到要求，甚至会使系统不能稳定运行。

要整定 PID 控制器的参数，需要做闭环实验。本书介绍的 PID 控制器的闭环仿真实验用 PLCSIM 对闭环系统仿真，用 WinCC flexible 的趋势图显示 PID 控制器的方波给定曲线和被控量的阶跃响应曲线。PLCSIM 不支持对 SFB41～SFB43 的仿真，但是可以对 FB41～FB43 仿真。

1．闭环仿真系统的结构

随书光盘中的例程“PID 控制”用功能块 FB100 模拟实际的执行机构和被控对象，与 FB41 组成闭环（见图 9-4）。不需要任何 PLC 的硬件和 PLC 外部的执行机构、检测元件和被控对象，就可以模拟闭环控制系统。

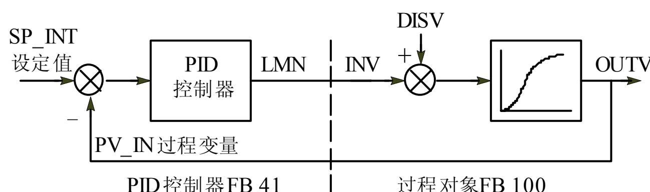


图 9-4　闭环仿真系统的结构

可以用人机界面画面上的 IO 域（见图 9-10）来修改 PID 控制器和被控对象的参数，通过运行用户程序，观察闭环控制的效果。人机界面的功能也是用仿真实现的。该例程可以用来学习 PID 控制功能块 FB41 的使用方法，熟悉 PID 控制器的参数的整定方法。

例程“PID 控制”中的闭环系统用 FB100 来模拟 3 个串联的惯性环节，其比例增益为 GAIN，惯性环节的时间常数分别为 TM_LAG1～TM_LAG3。其传递函数为

$$\frac{\mathrm{GAIN}}{(\mathrm{TM_LAG1}\,s+1)(\mathrm{TM_LAG2}s+1)(\mathrm{TM_LAG3}s+1)}$$

分母中的“s”为自动控制理论中拉式变换的拉普拉斯算子。将某一时间常数设置为 0，可以减少被控对象的阶数。图 9-4 中的 DISV 是系统的扰动量。

2．生成项目与站点

打开 STEP 7，用“新建项目向导”生成一个名为“PID 控制”的项目（见随书光盘中的同名例程），CPU 的型号为 CPU 315-2DP。

选中管理器左侧的项目窗口最上面的项目图标，执行菜单命令“插入”→“站点”→“SIMATIC HMI Station”，在出现的对话框中设置 HMI 的型号为 8in 的 MP277 触摸屏。

3．组态站点之间的连接

单击工具栏上的按钮，打开网络组态工具 NetPro（见图 8-6 上面的图），将 S7-300 站点和 HMI 站点连接到 MPI 网络上，它们的 MPI 站地址分别为 2 和 1。单击工具栏上的按钮，保存和编译网络组态信息。

4．打开 WinCC flexible 的项目

关闭 NetPro，返回 SIMATIC 管理器，选中 WinCC flexible 站点中的“画面”（见图 8-5），双击管理器右侧的工作区中的“画面_1”图标，自动打开 WinCC flexible 的“画面”编辑器。

双击 WinCC flexible 右侧项目视图中的“连接”，打开连接编辑器（见图 8-10），在“激活的”列选中“开”。

5．生成趋势图

打开“画面_1”，选中工具箱中的“趋势视图”，将它拖放到画面上（见图 9-10）。图 9-5～图 9-9 给出了组态趋势视图各参数组的属性视图。

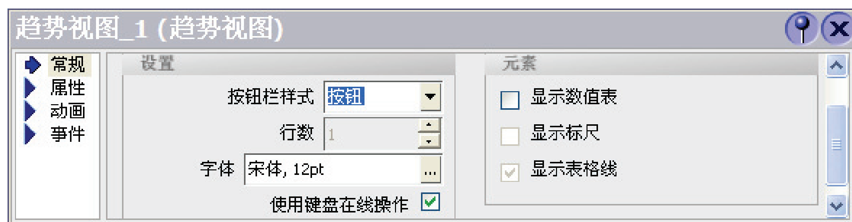


图 9-5　组态趋势视图的常规属性



图 9-6　组态趋势视图的 X 轴

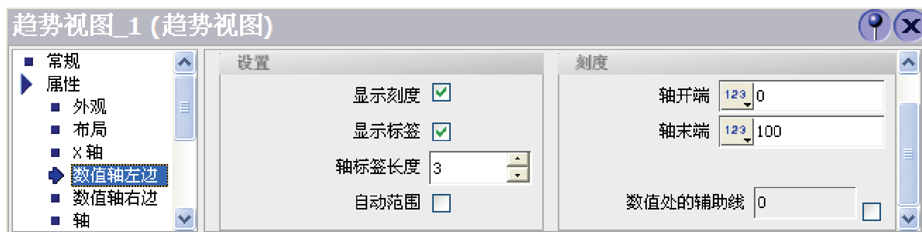


图 9-7　组态趋势视图左边的数值轴

组态右边数值轴时，未选中复选框“显示刻度”和“显示标签”，因此不会显示右边的数值轴。

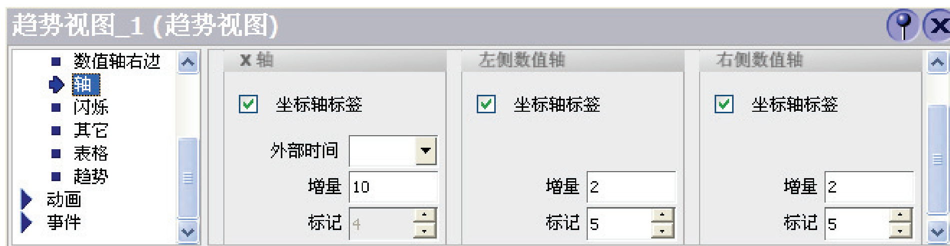


图 9-8　组态趋势图视图轴的刻度和显示的数字





图 9-9　组态趋势视图的趋势曲线

趋势视图用绿色实线显示给定值曲线，用黑色实线显示系统的输出变量的曲线。

6. 生成文本域和 IO 域

将工具箱中标有“A”的文本域图标（见图 8-9）拖放到画面上。选中生成的文本域，打开属性视图中的“常规”对话框，在右边的文本框中输入文本“比例系数”（见图 9-10）。

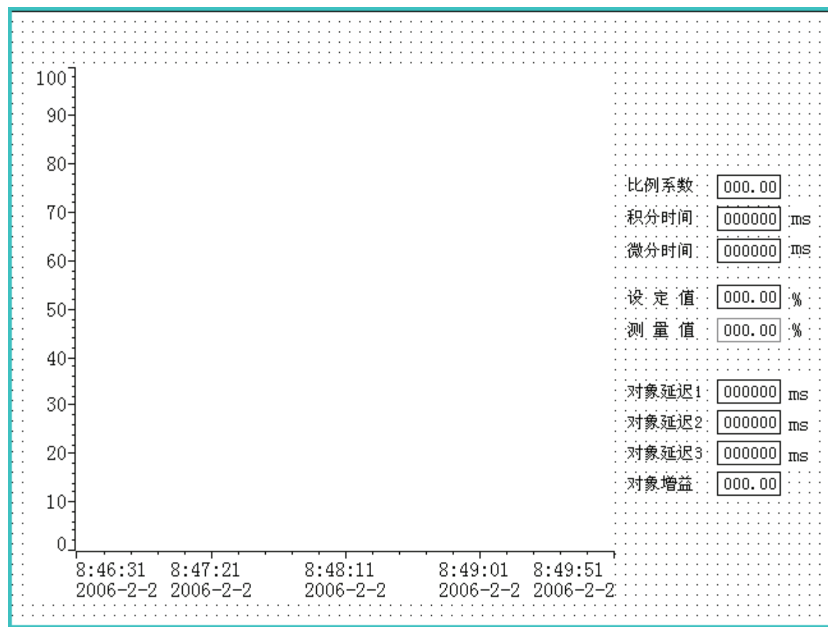


图 9-10　调试 PID 参数的画面

将工具箱中的 IO 域图标拖放到画面上“比例系数”文本域的右边，选中生成的 IO 域，在属性视图的“常规”对话框（见图 8-21），设置 IO 域为输入/输出域，连接的变量为“PID_DI.GAIN”（PID 控制器的增益），格式为默认的“十进制”，“格式样式”为 999.99（3 位整数、2 位小数）。在“属性”的“外观”对话框，设置边框的样式为“实心的”（有边框）。

用复制和粘贴操作，生成下列文本域和 IO 域（见图 9-10）：

1）文本域“设定值”和它右边用于显示 PID 控制器设定值 SP_INT 的输出域，其单位为百分数。

2）文本域“测量值”和它右边用于显示过程变量 PV_IN 的输出域，其单位为百分数。

3）文本域“对象增益”和它右边用于修改被控对象增益 GAIN 的输入/输出域。

在画面上生成文本域“积分时间”和它右边用于修改积分时间 TI 的输入/输出域，格式为默认的“十进制”，“格式样式”为 999999（6 位整数，单位为 ms），有边框。在 IO 域的右边，生成文本域“ms”。

用复制和粘贴操作，生成输入微分时间和对象延迟时间的输入/输出域，以及对应的文本域。

9.2.2 PID 控制程序

1. 示例程序的结构

打开硬件组态工具 HW Config，双击机架中的 CPU，打开 CPU 属性对话框。在“周期性中断”选项卡，将执行循环中断组织块 OB35 的时间间隔设置为 200ms。保存和编译组态信息后，将组态数据下载到仿真 CPU。程序运行时每隔 200ms 自动调用一次 OB35。

刚进入 RUN 模式时，CPU 执行一次 OB100，在 OB100 和 OB35 中调用 FB41 和 FB100，其背景数据块分别为 DB41 和 DB100。各逻辑块和数据块的符号名见图 9-11 中的符号表。

符号编辑器 - [S7 程序(4) (符号) -- PID控制\SIM...

符号表(S) 编辑(E) 插入(I) 视图(V) 选项(O) 窗口(W) 帮助(H)

全部符号

	状态	符号	地址		数据类型		注释
1		PID_DI	DB	41	FB	41	PID背景数据块
2		对象DI	DB	100	FB	100	模拟对象背景数据块
3		CONT_C	FB	41	FB	41	连续PID控制器
4		过程对象	FB	100	FB	100	模拟对象
5		循环中断	OB	35	OB	35	循环中断组织块
6		起动OB	OB	100	OB	100	起动组织块

图 9-11 符号表

2. 在 OB1 中产生方波给定信号

在 OB1 中，用 T8 和 T9 组成振荡电路（见图 9-12），T8 的常开触点的接通和断开的时间均为 40s。"PID_DI".SP_INT 是 FB41 的背景数据块 PID_DI 中存放控制器浮点数设定值的输入变量 SP_INT。在 T8 的常开触点刚接通和刚断开时，分别将设定值 SP_INT 修改为浮点数 20.0%和 70.0%，设定值的波形为方波。

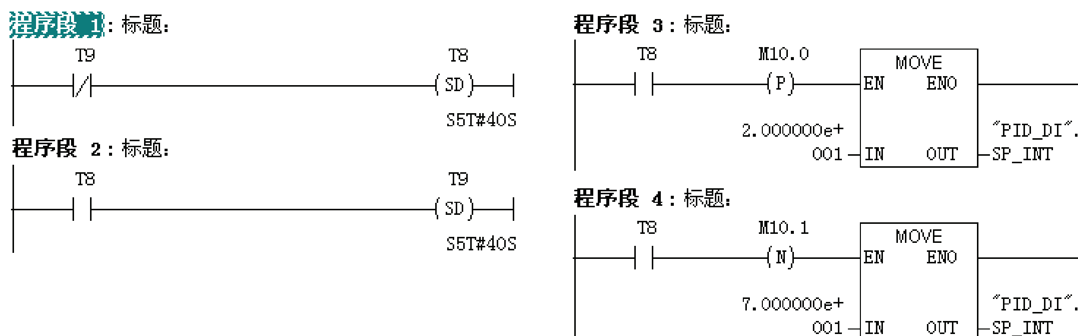


图 9-12 OB1 中的梯形图



3．起动组织块 OB100

功能块中局部变量的初始值是该变量的默认值。如果在调用功能块时没有给输入、输出参数指定实参，该参数的值为初始值。

下面是 OB100 中的程序，在调用 FB41 和 FB100 时，只设置了少量的实参：

1）令起动标志 COM_RST 为 TRUE，执行 PID 重启动功能，将 PID 控制器的内部参数设置为默认值。

2）设置采样周期 CYCLE 为 200ms，FB100 和 FB41 中的采样周期应与 OB35 的循环执行周期相同。

3）设置 PID 控制器的参数 GAIN、TI 和 TD 的初值。假设 AO 模块为双极性输出，设置控制器输出下限值为 -100.0%，上限值为默认的 100.0%。

```
程序段 1：调用连续 PID 控制器 FB41
      CALL   "CONT_C" , "PID_DI"
       COM_RST    :=TRUE                 //执行初始化程序
       MAN_ON     :=
       PVPER_ON   :=
       P_SEL      :=
       I_SEL      :=
       INT_HOLD   :=
       I_ITL_ON   :=
       D_SEL      :=
       CYCLE      :=T#200MS              //采样时间
       SP_INT     :=
       PV_IN      :=
       PV_PER     :=
       MAN        :=
       GAIN       :=2.000000e+000        //比例系数初值
       TI         :=T#2S                 //积分时间初值
       TD         :=T#800MS              //微分时间初值
       TM_LAG     :=
       DEADB_W    :=
       LMN_HLM    :=
       LMN_LLM    := -1.000000e+002      //控制器输出下限值
       PV_FAC     :=
       PV_OFF     :=
       LMN_FAC    :=
       LMN_OFF    :=
       I_ITLVAL   :=
       DISV       :=
       LMN        :=
       LMN_PER    :=
       QLMN_HLM:=
       QLMN_LLM:=
       LMN_P      :=
       LMN_I      :=
```

```
        LMN_D       :=
        PV          :=
        ER          :=
```

下面是调用模拟被控对象的功能块 FB100 的程序，设置了以下输入参数：

1）令起动标志 COM_RST 为 TRUE，初始化 FB100 的输出值和内部变量。

2）设置采样周期 CYCLE 为 200ms。

3）在 FB100 的局部变量表中，设置增益 GAIN 的初始值为 3.0，3 个惯性环节的时间常数（TM_LAG1～TM_TAG3）初始值分别为 5s、2s 和 0s。

```
程序段 2：初始化模拟被控对象的 FB 100
      CALL   "过程对象" , "对象 DI"            //调用 FB 100
        INV         :=
        DISV        :=
        GAIN        :=
        TM_LAG1     :=
        TM_LAG2     :=
        TM_LAG3     :=
        COM_RST     :=TRUE                    //启动标志
        CYCLE       :=T#200MS                 //采样周期
        OUTV        :=
```

在退出 OB100 之前，用下面的程序将 FB41 和 FB100 的启动标志位 COM_RST 复位，将 PID 控制器的给定值设置为 70.0%。

```
程序段 3：复位启动标志
      CLR
      =       "PID_DI".COM_RST
      =       "对象 DI".COM_RST
程序段 4：令 PID 给定值的初值为 70%
      L       7.000000e+001
      T       "PID_DI".SP_INT
```

4．循环中断组织块 OB35

为了保证 PID 运算的采样周期的精度，用循环中断组织块 OB35 调用 FB41 和 FB100。

在 OB35 中，将会使用在 OB100 中设置的 FB41 和 FB100 的参数，直到用人机界面的 IO 域修改其中的某些参数。在 OB100 和 OB35 中未设置的 FB41 和 FB100 的输入、输出参数，将使用它们的初始值。在 FB41 和 FB100 的局部变量表或在线帮助中，可以看到局部变量的初始值。

FB41 的背景数据块 DB41 的符号名为“PID_ID”，FB100 的背景数据块 DB100 的符号名为“对象 DI”。将 PID 控制器的输出变量 LMN 送给 FB100 的输入变量 INV，将 FB100 的输出变量 OUTV 送给 PID 控制器的过程输入变量 PV_IN，FB41 和 FB100 组成了图 9-4 所示的闭环。因为是用软件对被控对象仿真，没有使用模拟量输入模块和模拟量输出模块，闭环中的变量均为浮点数。

FB41 默认的设置为 PI 控制器，将参数 D_SEL 设置为 1（TRUE），该控制器为 PID 控



制器。设置 MAN_ON 为 FALSE，控制器工作在自动方式。

下面的程序段 2 的指令用于计算微分操作的延迟时间 TM_LAG，其推荐值为 TD/5。

输入端的选择开关 PVPER_ON 为默认值 FALSE，采用浮点数格式的过程变量。过程变量不是来自外设（AI 模块），不用设置 PV_PER（外设输入过程值）。因为 PID 的输出不是送给 AO 模块，不用设置 I/O 格式的控制器输出 LMN_PER 的实参。

在实际的语句表程序中，如果要给功能块的某个输入、输出变量加注释，必须设置该变量的实参。下面是循环中断组织块 OB35 的程序。

```
程序段 1: 调用被控对象仿真程序 FB100
      CALL   "过程对象" , "对象 DI"
       INV          :="PID_DI".LMN      //PID 控制器的浮点数输出值作为被控对象的输入变量
       DISV         :=                  //扰动量，初始值为 0.0
       GAIN         :=                  //比例系数，初始值为 3.0，可用触摸屏修改
       TM_LAG1      :=                  //时间常数 1，初始值为 5s，可用触摸屏修改
       TM_LAG2      :=                  //时间常数 2，初始值为 2s，可用触摸屏修改
       TM_LAG3      :=                  //时间常数 3，初始值为 0s，可用触摸屏修改
       COM_RST      :=                  //启动标志，在 OB100 被复位
       CYCLE        :=                  //采样周期，在 OB100 被置为 T#200MS
       OUTV         :="PID_DI".PV_IN    //浮点数输出，作为 PID 控制器的过程变量（反馈值）

程序段 2: 计算微分操作的延迟时间，TM_LAG = TD/5
      L      "PID_DI".TD
      L      5.000000e+000
      /R
      T      "PID_DI".TM_LAG

程序段 3: 调用连续 PID 控制器 FB41
      CALL   "CONT_C" , "PID_DI"
       COM_RST      :=                  //启动标志，在 OB100 被复位
       MAN_ON       :=FALSE             //自动运行
       PVPER_ON     :=                  //采用默认值 FALSE，使用浮点数过程值
       P_SEL        :=                  //采用默认值 TRUE，打开比例（P）操作
       I_SEL        :=                  //采用默认值 TRUE，打开积分（I）操作
       INT_HOLD     :=                  //采用默认值 FALSE，不冻结积分输出
       I_ITL_ON     :=                  //采用默认值 FALSE，不设积分器的初值
       D_SEL        :=TRUE              //打开微分（D）操作，默认值为 FALSE
       CYCLE        :=                  //采样周期，在 OB100 被置为 T#200MS
       SP_INT       :=                  //在 OB1 中修改此设定值
       PV_IN        :="对象 DI".OUTV    //来自 FB100 的浮点数格式的过程变量输入
       PV_PER       :=                  //外部设备输入的 I/O 格式的过程变量值，未用
       MAN          :=                  //手动值，未用
       GAIN         :=                  //比例系数，在 OB100 被初始化为 2.0，可用触摸屏修改
       TI           :=                  //积分时间，在 OB100 被初始化为 2s，可用触摸屏修改
       TD           :=                  //微分时间，在 OB100 被初始化为 0.8s，可用触摸屏修改
```

```
TM_LAG      :=                  //微分部分的延迟时间，微分时间 TD 的 1/5
DEADB_W     :=                  //死区宽度，默认值为 0.0（无死区）
LMN_HLM     :=                  //控制器输出上限值，默认值为 100.0
LMN_LLM     :=                  //控制器输出下限值，初值为 -100.0
PV_FAC      :=                  //输入的过程变量的系数，默认值为 1.0
PV_OFF      :=                  //输入的过程变量的偏移量，默认值为 0.0
LMN_FAC     :=                  //控制器输出量的系数，默认值为 1.0
LMN_OFF     :=                  //控制器输出量的偏移量，默认值为 0.0
I_ITLVAL    :=                  //积分操作的初始值，未用
DISV        :=                  //扰动输入变量，默认值为 0.0
LMN         :="对象 DI".INV     //控制器浮点数输出值送被控对象
LMN_PER     :=                  //I/O 格式的控制器输出值地址，未用
QLMN_HLM:=                      //控制器输出超过上限，未用
QLMN_LLM:=                      //控制器输出小于下限，未用
LMN_P       :=                  //控制器输出值中的比例分量，未用
LMN_I       :=                  //控制器输出值中的积分分量，未用
LMN_D       :=                  //控制器输出值中的微分分量，未用
PV          :=                  //格式化的过程变量，未用
ER          :=                  //死区处理后的误差，未用
```

9.2.3 PID 控制器的参数整定方法

PID 控制器的参数与系统动态性能和稳态性能之间的关系是参数整定的基础。

控制器输出的比例、积分、微分部分都有明确的物理意义，在整定 PID 控制器参数时，可以根据控制器的参数与系统动态性能和稳态性能之间的定性关系，用实验的方法来调节控制器的参数。有经验的调试人员一般可以较快地得到较为满意的调试结果。

在调试中最重要的问题是在系统性能不能令人满意时，知道应该调节哪一个参数，该参数应该增大还是减小。

PID 的控制原理可以用人对炉温的手动控制来理解。假设用热电偶检测炉温，用数字显示温度值。

1．比例控制

在控制过程中，操作人员用眼睛读取炉温，并与炉温的给定值比较，得到温度的误差值。用手操作电位器，调节加热的电流，使炉温保持在给定值附近。

操作人员知道炉温稳定在给定值时电位器的大致位置（我们将它称为位置 L），并根据当时的温度误差值调整控制加热电流的电位器的转角。炉温小于给定值时，误差为正，在位置 L 的基础上顺时针增大电位器的转角，以增大加热的电流。炉温大于给定值时，误差为负，在位置 L 的基础上反时针减小电位器的转角，并令转角与位置 L 的差值与误差成正比。上述控制策略就是比例控制，即 PID 控制器输出中的比例部分与误差成正比，比例系数为 FB41 的输入参数增益 GAIN。

闭环中存在着各种各样的延迟作用。例如调节电位器转角后，到温度上升到新的转角对应的稳态值时有较大的延迟。温度的检测、转换为数字量和 PID 的周期性计算都有延迟。由于延迟因素的存在，调节电位器转角后不能马上看到调节的效果，因此闭环控制系统调节困



难的主要原因是系统中的延迟作用。

如果比例系数太小，即调节后电位器转角与位置 L 的差值太小，调节的力度不够，使系统输出量变化缓慢，调节时间过长。比例系数过大（即调节后电位器转角与位置 L 的差值过大），调节力度太强，造成调节过头，甚至使温度忽高忽低，来回震荡。

如果闭环系统没有积分作用，由理论分析可知，单纯的比例控制有稳态误差，稳态误差与比例系数成反比。增大比例系数使系统反应灵敏，调节速度加快，并且可以减小稳态误差。但是比例系数过大会使超调量增大，振荡次数增加，调节时间加长，导致动态性能变坏，比例系数过大甚至会使闭环系统不稳定。

2．积分控制

式（9-1）中的积分对应于图 9-13 中曲线 $ev(t)$ 与坐标轴包围的面积（见图中的灰色部分），一般用图中矩形面积之和来近似精确积分。即每次 PID 运算时，在原来的积分值的基础上，增加一个与当前的误差值成正比的微小部分。误差为负值时，积分的增量为负。

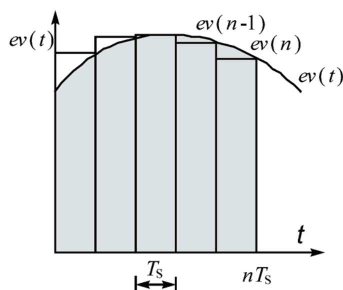


图 9-13　积分的近似运算

在上述的温度控制系统中，积分控制相当于根据当时的误差值，周期性地微调电位器的角度。温度低于设定值时误差为正，积分项增大，使加热电流增加，反之积分项减小。因此只要误差不为零，控制器的输出就会因为积分作用而不断变化。积分调节的“大方向”是正确的，积分项有减小误差的作用，一直要到系统处于稳定状态，这时误差恒为零，比例部分和微分部分均为零，积分部分才不再变化，并且刚好等于稳态时需要的控制器的输出值，对应于上述温度控制系统中电位器转角的位置 L。因此积分部分的作用是消除稳态误差，提高控制精度，积分作用一般是必须的。

积分部分与误差的积分成正比。因为积分时间 T_I 在式（9-1）的积分项的分母中，T_I 越小，积分项变化的速度越快，积分作用越强。

积分项根据当前误差值和过去的历次误差值累加而成，因此积分作用本身具有严重的滞后特性，对系统的稳定性不利。如果积分项的系数设置得不好，其负面作用很难通过积分作用本身迅速地修正。

如果积分作用太强，相当于每次微调电位器的角度值过大，其累积的作用会使系统输出的动态性能变差，超调量增大，甚至使系统不稳定。积分作用太弱，则消除稳态误差的速度太慢。

PID 的比例部分没有延迟，只要误差一出现，比例部分就会立即起作用。因此积分作用很少单独使用，它一般与比例和微分部分联合使用，组成 PI 或 PID 控制器。

PI 和 PID 控制器既克服了单纯的比例调节有稳态误差的缺点，又避免了单纯的积分调节响应慢、动态性能不好的缺点，因此被广泛使用。

如果 PID 控制器有积分作用（例如采用 PI 或 PID 控制），积分能消除阶跃输入的稳态误差，这时可以将比例系数调得小一些。

综上所述，积分作用太强（即 T_I 太小），系统的稳定性变差，超调量增大；积分作用太弱（即 T_I 太大），系统消除稳态误差的速度太慢，T_I 的值应取得适中。

3．微分作用

误差的微分就是误差的变化速率，误差变化越快，其微分绝对值越大。误差增大时，其微分为正；误差减小时，其微分为负。控制器输出量的微分部分与误差的微分成正比，反映了被控量变化的趋势。

有经验的操作人员在温度上升过快，但是尚未达到设定值时，根据温度变化的趋势，预感到温度将会超过设定值，出现超调。于是调节电位器的转角，提前减小加热的电流。这相当于士兵射击远方的移动目标时，考虑到子弹运动的时间，需要一定的提前量一样。

在图 9-2 中启动过程的上升阶段，当 $c(t) < c(\infty)$ 时，被控量尚未超过其稳态值，超调量还没有出现。但是因为误差 $e(t)$ 不断减小，误差的微分和控制器输出的微分部分为负，减小了控制器的输出量，相当于提前给出了制动作用，以阻碍被控量的上升，所以可以减少超调量。因此微分控制具有超前和预测的特性，在超调尚未出现之前，就能提前给出控制作用。

闭环控制系统的振荡甚至不稳定的根本原因在于有较大的滞后因素，因为微分项能预测误差变化的趋势，这种“超前”的作用可以抵消滞后因素的影响，适当的微分控制作用可以使超调量减小，调节时间缩短，增加系统的稳定性。其缺点是对干扰噪声敏感，使系统抑制干扰的能力降低。

对于有较大惯性或滞后的被控对象，控制器输出量变化后，要经过较长的时间才能引起反馈量的变化。如果 PI 控制器的控制效果不理想，可以考虑在控制器中增加微分作用，以改善系统在调节过程中的动态特性。

微分时间 T_D 与微分作用的强弱成正比，T_D 越大，微分作用越强。但是 T_D 太大，对误差变化率的压抑过度，可能会使响应曲线上出现“毛刺”（见图 9-16），或使被控量接近稳态值时变化缓慢。后一现象的原因是因为接近稳态值时，误差很小，比例部分消除误差的能力很弱。因为微分部分太强，抑制了被控量的上升，导致被控量上升极为缓慢，到达稳态的时间过长。如果将微分时间设置为 0，微分部分将不起作用。

4．采样周期

PID 控制程序是周期性执行的，执行的周期称为采样周期。采样周期 T_S 越小，采样值越能反映模拟量的变化情况。但是 T_S 太小会增加 CPU 的运算工作量，相邻两次采样的差值几乎没有什么变化，将使 PID 控制器输出的微分部分接近为零，所以也不宜将 T_S 取得过小。

确定采样周期时，应保证在被控量迅速变化时（例如启动过程中的上升阶段），能有足够多的采样点数，以保证不会因为采样点过稀而丢失被采集的模拟量中的重要信息。

表 9-1 给出了过程控制中采样周期的经验数据，表中的数据仅供参考，实际的采样周期需要经过现场调试后确定。

表 9-1　采样周期的经验数据

被控制量	流　量	压　力	温　度	液　位	成　份
采样周期/s	1～5	3～10	15～20	6～8	15～20

5．PID 参数的调整方法

为了减少需要整定的参数，可以首先采用 PI 控制器。为了保证系统的安全，避免出现



系统不稳定或超调量过大的异常情况，在调试开始时应设置比较保守的参数，例如增益不要太大，积分时间不要太小。给出一个阶跃给定信号，观察系统输出量的波形。由输出波形可以获得系统性能的信息，例如超调量和调节时间。根据 PID 参数与系统性能的关系，反复调节 PID 的参数。

如果阶跃响应的超调量太大，经过多次振荡才能稳定或者根本不稳定，应减小 PID 输出的比例系数、增大积分时间。如果阶跃响应没有超调量，但是被控量上升过于缓慢，过渡过程时间太长，应按相反的方向调整参数。

如果消除误差的速度较慢，可以适当减小积分时间。

反复调节 K_P 和 T_I，如果超调量仍然较大，可以加入微分，T_D 从 0 逐渐增大，反复调节 K_P、T_I 和 T_D。

需要注意的是在改变增益 K_P 的值时，同时会影响到积分分量和微分分量的值（见式 9-1），而不是仅仅影响到比例分量。

总之，PID 参数的调试是一个综合的、各参数互相影响的过程，实际调试过程中的多次尝试是非常重要的，也是必须的。

9.2.4　PID 控制的仿真实验

1．起动 PLCSIM

在 SIMATIC 管理器中，打开项目“PID 控制”。单击工具栏上的按钮，打开 PLCSIM。将用户程序和组态信息下载到仿真 PLC，将仿真 PLC 切换到 RUN-P 模式。

2．用 STEP 7 的变量表监控程序的运行

根据调试的需要，可以用变量表集中监视和修改 PID 控制器的关键参数。图 9-14 是调试 PID 控制例程时使用的变量表，单击按钮，“修改数值”被传送到 PLC。

变量 - [VAT1 -- @PID控制\SIMATIC 300 站点\CPU 315-2 DP\...

表格(T)　编辑(E)　插入(I)　PLC　变量(A)　视图(V)　选项(O)　窗口(W)　帮助(H)

	地址	符号	显示格式	状态值	修改数值
1	DB41.DBD 20	"PID_DI".GAIN	FLOATING_POINT	2.0	
2	DB41.DBD 24	"PID_DI".TI	TIME	T#2s	
3	DB41.DBD 28	"PID_DI".TD	TIME	T#800ms	
4	DB41.DBD 6	"PID_DI".SP_INT	FLOATING_POINT	20.0	
5	DB41.DBD 10	"PID_DI".PV_IN	FLOATING_POINT	4.384813	
6	DB41.DBD 72	"PID_DI".LMN	FLOATING_POINT	18.43293	
7	DB41.DBD 96	"PID_DI".ER	FLOATING_POINT	15.61519	
8	DB100.DBD 12	"对象DI".TM_LAG1	TIME	T#5s	
9	DB100.DBD 16	"对象DI".TM_LAG2	TIME	T#2s	
10	DB100.DBD 20	"对象DI".TM_LAG3	TIME	T#0ms	
11	DB100.DBD 8	"对象DI".GAIN	FLOATING_POINT	3.0	

图 9-14　变量表

在运行时监视闭环控制系统的输出量 OUTV（见图 9-4），或 FB41 的过程变量输入值 PV_IN（二者相等），可以观察到 PID 控制的效果。

通过监视功能块的背景数据块，可以监视功能块除了临时局部变量之外的其他变量，还

可以看到调用功能块时未指定实参的变量的默认值。

上述监控方法不像下面将要看到的人机界面趋势视图中的曲线那样形象直观，如果系统的动态过程变化太快，监控更为困难。

3．用触摸屏的趋势视图监控 PID 控制程序

打开 WinCC Flexible，单击工具栏上的按钮，起动运行系统，出现仿真面板（见图 9-15）。

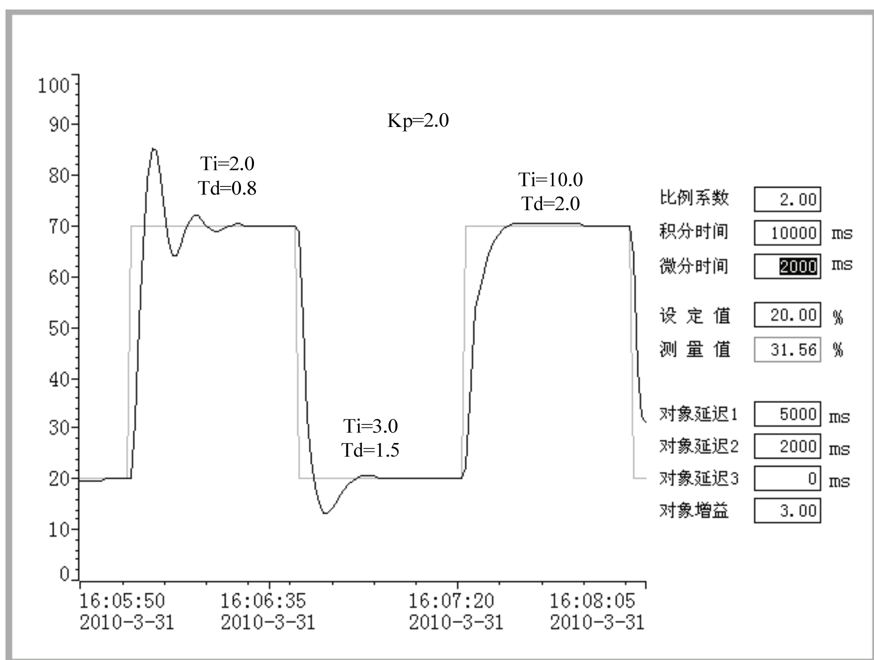


图 9-15　PID 的阶跃响应曲线

由于 PLC 的 OB1 中梯形图程序的作用，PID 控制器的给定值在 20%～70%之间阶跃变化，趋势视图中对应的曲线为绿色的方波。被控对象的输出变量（即 PID 控制器的过程变量或反馈值）为黑色的曲线。

下面是 PID 控制器的初始值：比例系数为 2.00，积分时间为 2.0s，微分时间为 0.8s，对象的增益和延迟时间见图 9-15。图中左边的给定输入的上升沿对应的响应曲线使用的是 PID 控制器参数的初始值，该曲线的超调量过大，有多次震荡。

单击画面上的文本域“积分时间”右侧的输入/输出域（见图 9-15），用出现的数字键盘将积分时间改为 3000ms。用同样的方法将微分时间改为 1500ms。中间的下降曲线是修改参数后的响应曲线，其超调量和震荡次数显著减小。

将积分时间改为 10000ms，微分时间改为 2000ms。右边的上升曲线是对应的响应曲线，其超调量几乎为 0。

图 9-16 的积分时间为 3000ms，微分时间为 4000ms。因为微分作用过强，在误差剧烈变化时，对误差变化的抑制作用太剧烈，曲线上出现了“毛刺”，变得很怪异。这种现象提示应减弱微分部分。



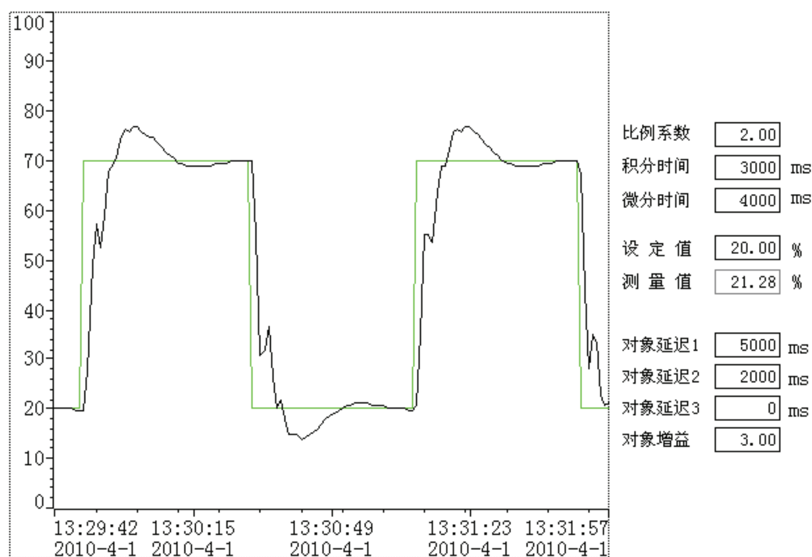


图 9-16　微分作用过强的阶跃响应曲线

图 9-17 的微分时间恒为 0（即采用 PI 调节），积分时间为 8000ms。左边和中间的响应曲线的控制器增益（即比例系数）为 2.0，右边的响应曲线的控制器增益为 1.0，比例作用和积分作用被同时减弱。可以看出，减小控制器增益能显著降低超调量。

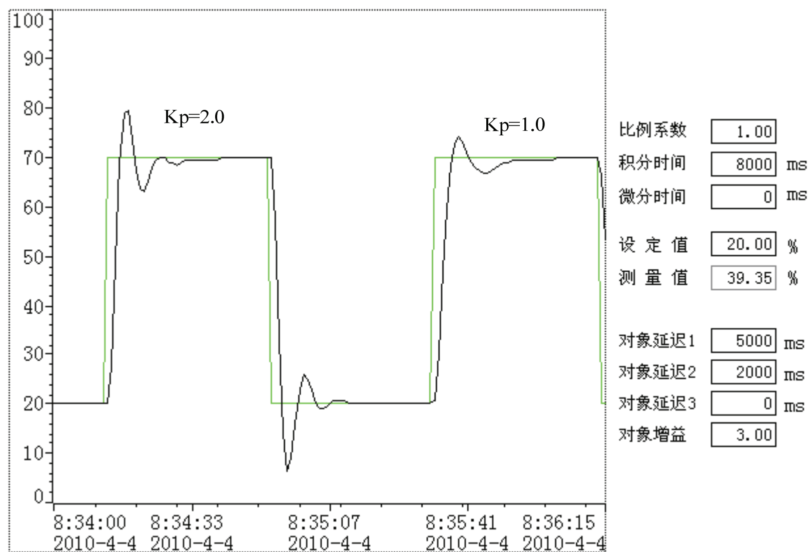


图 9-17　PI 调节器的阶跃响应曲线

4．PID 参数整定的仿真练习

读者可以利用随书光盘中的项目“PID 控制”，进行下面的仿真练习：

1）运行仿真闭环控制系统，修改图 9-15 中 PID 控制器的比例系数、积分时间和微分时间，观察控制器参数与系统性能之间的关系，学习 PID 参数的整定方法。

2）在图 9-15 中 PID 的参数固定不变的情况下，改变 OB35 的循环执行周期（即 PID 控

制器的采样周期）和 FB41、FB100 中的参数 CYCLE（三者应相同），观察采样周期对系统性能的影响，了解整定采样周期的方法。

3）令微分时间恒为 0（即采用 PI 调节），调节比例系数和积分时间，直到得到较为满意的控制效果。

4）修改被控对象的参数，然后调节 PID 控制器的参数，直到得到较为满意的控制效果。

5．仿真系统的程序与实际的 PID 程序的区别

为了用于实际的 PID 控制，在例程“PID 控制”程序的基础上，应作下列改动：

1）删除 OB100 和 OB35 中调用 FB100（过程对象）的指令。

2）实际的 PID 控制程序使用来自 AI 模块的过程变量 PV_PER，用来选择输入变量的 PVPER_ON 应设置为 TRUE（使用外设变量），不用设置浮点数过程变量 PV_IN，外设变量 PV_PER 应设为实际使用的模拟量输入模块的通道地址。

3）不用设置浮点数输出 LMN 的实参，LMN_PER（外设输出值）设为实际使用的模拟量输出模块的通道地址。

4）如果系统需要自动/手动两种工作模式的切换，FB41 的参数 MAN_ON 应设置为切换自动/手动的位变量，手动时该变量为 ON。参数 MAN 设置为用于输入手动值的地址。

9.3 练习题

1．简述闭环控制的工作原理。

2．超调量反映了闭环系统的什么性能？

3．死区在闭环控制中有什么作用？

4．微分操作的延迟时间 TM_LAG 有什么作用？

5．简述使用 FB41 实现 PID 控制的程序结构。

6．怎样用 FB41 实现 PI 或 PID 控制？

7．超调量大应怎样调节 PID 控制器的参数？

8．被控量过渡过程过于缓慢应怎样调节 PID 控制器的参数？

9．消除误差的速度较慢应怎样调节 PID 控制器的参数？

10．PID 的微分部分有什么作用？微分作用过强有什么负面影响？

附录

随书光盘内容简介

光盘中后缀为 pdf 的用户手册需要用 Adobe reader（Adobe 阅读器）阅读，可以在安装 STEP 7 时选择安装 Adobe Reader，或在网上下载该软件的最新版本。

1．软件

1）包含 SP3.1 的 STEP 7 V5.4 中文版。

2）S7-PLCSIM V5.4。

3）S7-PLCSIM V5.4 SP3。

4）S7 GRAPH V5.3。

5）WinCC flexible 2008。

6）PC_Adapter_USB：USB/MPI 适配器的驱动程序。

2．多媒体视频教程

生成项目与组态硬件

生成与调试用户程序

生成与调用功能块

专用钻床顺序控制与 SIMIT 被控对象仿真

DP 从站组态与故障诊断

整定 PID 参数的仿真实验

3．软件手册

PLCSIM V5.4 SP3 Operating Manual。

PLCSIM 使用入门。

S7 PID 控制用户手册。

S7-300_400 编程梯形图(LAD)参考手册。

S7-300_400 编程语句表(STL)参考手册。

S7-GRAPH V5.3 Manual。

使用 STEP 7 编程手册。

用于 S7 的系统软件和标准功能参考手册。

4．硬件手册

CPU 31xC 和 CPU 31x 技术规范设备手册。

CPU 31xC 和 CPU 31x 安装操作说明。

ET 200 产品目录。

S7-300 模块数据设备手册。

S7-300 产品目录。

S7-400 CPU 规格设备手册。
S7-400 模块数据参考手册。
S7-400 产品样本。
组态硬件和通信连接手册。

5．例程

\Project\电机控制：三相异步电动机正反转控制。
\Project\小车控制 1：小车在两个限位开关之间往返运动。
\Project\位逻辑：位逻辑指令的应用例程。
\Project\车库入口：位逻辑指令的应用例程。
\Project\定时器 1：5 种 S5 定时器指令的基本应用程序。
\Project\定时器 2：5 种定时器线圈指令的基本应用程序。
\Project\运输带控制：两条运输带和 3 条运输带的控制例程。
\Project\小车控制 2：用经验设计法设计的小车控制程序。
\Project\计数器：方框计数器指令和计数器线圈指令的应用程序。
\Project\FC 例程：功能应用例程。
\Project\FB 例程：功能块应用例程。
\Project\数组_SFC：共享数据块、数组和系统功能块应用例程。
\Project\多重背景：多重背景应用例程。
\Project\OB35 例程：使用 OB35 的例程。
\Project\OB10_1：用组态设置日期时间的时间中断例程。
\Project\OB10_2：用程序设置日期时间的时间中断例程。
\Project\OB40 例程：硬件中断例程。
\Project\OB20 例程：延时中断例程。
\Project\运输带顺控：两条运输带顺序控制例程。
\Project\钻床控制：专用钻床顺序控制例程。
\Project\运输带 GR：使用 S7-GRAPH 的运输带控制例程。
\Project\剪板机 GR：使用 S7-GRAPH 的剪板机控制例程。
\Project\ET200：ET 200 作从站的 PROFIBUS-DP 通信例程。
\Project\EM277：S7-200 作从站的 PROFIBUS-DP 通信例程。
\Project\智能从站：S7-300 作智能从站的 PROFIBUS-DP 通信例程。
\Project\SFC14_15：S7-300 作智能从站的 PROFIBUS-DP 通信例程，使用 SFC 14/15。
\Project\S7_DP：基于 DP 网络的单向 S7 通信例程。
\Project\S7_MPI：基于 MPI 网络的双向 S7 通信例程。
\Project\从站诊断：调用 OB86 的 DP 从站故障诊断例程。
\Project\机架故障：调用 OB86 的扩展机架故障诊断例程。
\Project\诊断中断：调用 OB82 的 I/O 模块故障诊断例程。
\Project\OB121：编程错误中断处理的例程。
\Project\PLC_HMI：PLC 与人机界面集成仿真例程。
\Project\PID 控制：PID 闭环控制系统的仿真例程。

参 考 文 献

[1] 廖常初. S7-300/400 PLC 应用技术[M]. 2 版. 北京：机械工业出版社，2008.

[2] 廖常初，陈晓东. 西门子人机界面（触摸屏）组态与应用技术[M]. 2 版. 北京：机械工业出版社，2008.

[3] 廖常初，祖正容. 西门子工业网络的组态编程与故障诊断[M]. 北京：机械工业出版社，2009.

[4] 廖常初. PLC 编程及应用[M]. 3 版. 北京：机械工业出版社，2008.

[5] 廖常初. S7-1200 PLC 编程及应用[M]. 2 版. 北京：机械工业出版社，2010.

[6] 廖常初. FX 系列 PLC 编程及应用[M]. 北京：机械工业出版社，2005.

[7] Siemens AG. Programming with STEP 7 Manual，2006.

[8] Siemens AG. System Software for S7-300/400 System and Standard Functions Reference Manual，2006.

[9] Siemens AG. Ladder Logic (LAD) for S7-300 and S7-400 Programming Reference Manual，2006.

[10] Siemens AG. Statement List (STL) for S7-300 and S7-400 Programming Reference Manual，2006.

[11] Siemens AG. S7-PLCSIM V5.4 incl. SP3 Operating Manual，2009.

[12] Siemens AG. PID Control User Manual，2003.

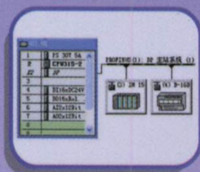
[13] Siemens AG. S7 GRAPH V5.3 for S7-300/400 Programming Sequential Control Systems Manual，2004.

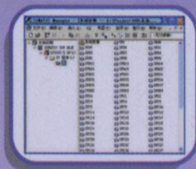
ISBN 978-7-111-31641-1

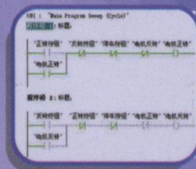
策　划：时 静

封面设计：子时文化 ZiShi Culture

跟我动手学 S7-300/400 PLC









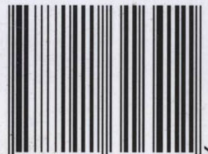
内容提要

S7-300/400是国内应用最广、市场占有率最高的大中型PLC。本书强调动手，强调实际操作。书中有四十多个实训，随书光盘有三十多个与正文配套的例程和多媒体视频教程。读者一边看书，一边根据实训的要求，用随书光盘的编程软件和仿真软件在计算机上做仿真实验，就能轻松上手，快速入门，掌握软件安装、硬件和网络组态、编程、监控、故障诊断、指令应用、程序结构、编程方法、通信、人机界面应用和PID控制等方面的知识和操作方法。并且会留下深刻的印象，不容易忘记。PID闭环仿真例程可以用来学习整定PID参数的方法。

本书各章配有适量的习题，可以供工程技术人员自学，也可以用来作高职高专、技工学校有关专业的教材。

上架建议　工业技术/电气工程

ISBN 978-7-111-31641-1



地址：北京市百万庄大街22号　邮政编码：100037
电话服务
社服务中心：(010)88361066
销 售 一 部：(010)68326294
销 售 二 部：(010)88379649
读者购书热线：(010)88379203
网络服务
门户网：http://www.cmpbook.com
教材网：http://www.cmpedu.com
封面无防伪标均为盗版

定价：32.00元（含1DVD）